Sommario

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Balocchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Segretaria di redazione: Maura Ceccaroli **Grafica:** Arturo Ciaglia, Gabriella Galbusera
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Guido Agosti, Giorgio Ruffoni, Claudio Tidone - v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Paola Bertolotti - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.500. Abbonamento annuo (11 fascicoli) L. 45.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 85.000. I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib.:** MePe - via G. Carcano, 32 - Milano
Periodi Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

I SEGRETI DEL COMPUTER

Crittografare, con la massima segretezza, informazioni riservate è il sogno dei potenti; ma non è solo un problema di natura tecnica

Di recente la stampa si occupa di presunti scandali legati a corruzioni e concussioni che hanno avuto, tra i protagonisti, anche un calcolatore.

Il problema della "sicurezza" dei dati va scisso in due tronconi, in apparente opposizione tra loro.

Da un lato c'è chi non ha nessuna intenzione di divulgare informazioni riservate che, per la loro mole, devono esser memorizzate ricorrendo a sistemi informatici (e qui ci riferiamo allo scambio di dati tra banche, agenzie legali, Polizia, Magistratura e così via). D'altra parte, invece, notiamo coloro che operano contro la legge e che utilizzano tecniche analoghe (il riferimento alla contabilità in nero è esplicito).

In entrambi i casi, dunque, vi sono dati da custodire con cura, che nessuno dovrebbe avere la possibilità di esaminare.

La realizzazione di procedure di sicurezza è da tempo in cima ai pensieri degli operatori che, per motivi opposti, ne inventano di tutte pur di ottenere la riservatezza desiderata.

Il problema della crittografia, a nostro parere, è semplicissimo da risolvere: perfino con un modesto home computer, e lavorando addirittura in Basic, è possibile realizzare archivi in codice, memorizzare, ben custodite, lettere compromettenti, nascondere tra le tracce di un dischetto i numeri di un assegno che scotta.

Chi lo desidera, pertanto, può realizzare centinaia di documenti segreti, da richiamare "in chiaro" digitando, al momento opportuno e con la massima semplicità, il codice segreto: questo può essere la propria data di nascita, il nome della prima morosa, un versetto di Dante, lo slogan della Coca Cola o il soprannome del nonno di un nostro amico; nessuno potrà mai decodificare il documento, se noi non lo vogliamo.

Come è possibile, dunque, che nomi, date, cifre possano uscire da un dischetto definito "in codice"?

La risposta è molto semplice e risiede nella stessa attività dell'operatore che intenderebbe ricorrere a codici segreti; vediamo di chiarire.

Un imprenditore, indaffarato nei suoi affari, non ha il tempo di studiare informatica, creare programmi e codici; mentre è in giro a cercare appalti e contattare autorità di alto livello è costretto, inevitabilmente, ad affidare la memorizzazione dei suoi affari ad un'altra persona che potrebbe, oppure no, capirne di computer.

Se, comunque, si è costretti a ricorrere ad un computer per memorizzare la cospicua contabilità (in nero, ovviamente) è perchè, appunto, gli "affari" sono talmente numerosi che non è possibile tenerli a mente o annotarli su un foglietto codificati in modo semplice ed elementare ("Per Nicola oggi ho comprato 12 rose rosse", e simili).

Se, inoltre, si deve memorizzare il tutto su un disco, è molto probabile che questo debba restare costantemente a disposizione, infilato, magari, tra dischi "puliti", per seguirne costantemente le vicende, proprio come se fosse la contabilità di un'altra azienda.

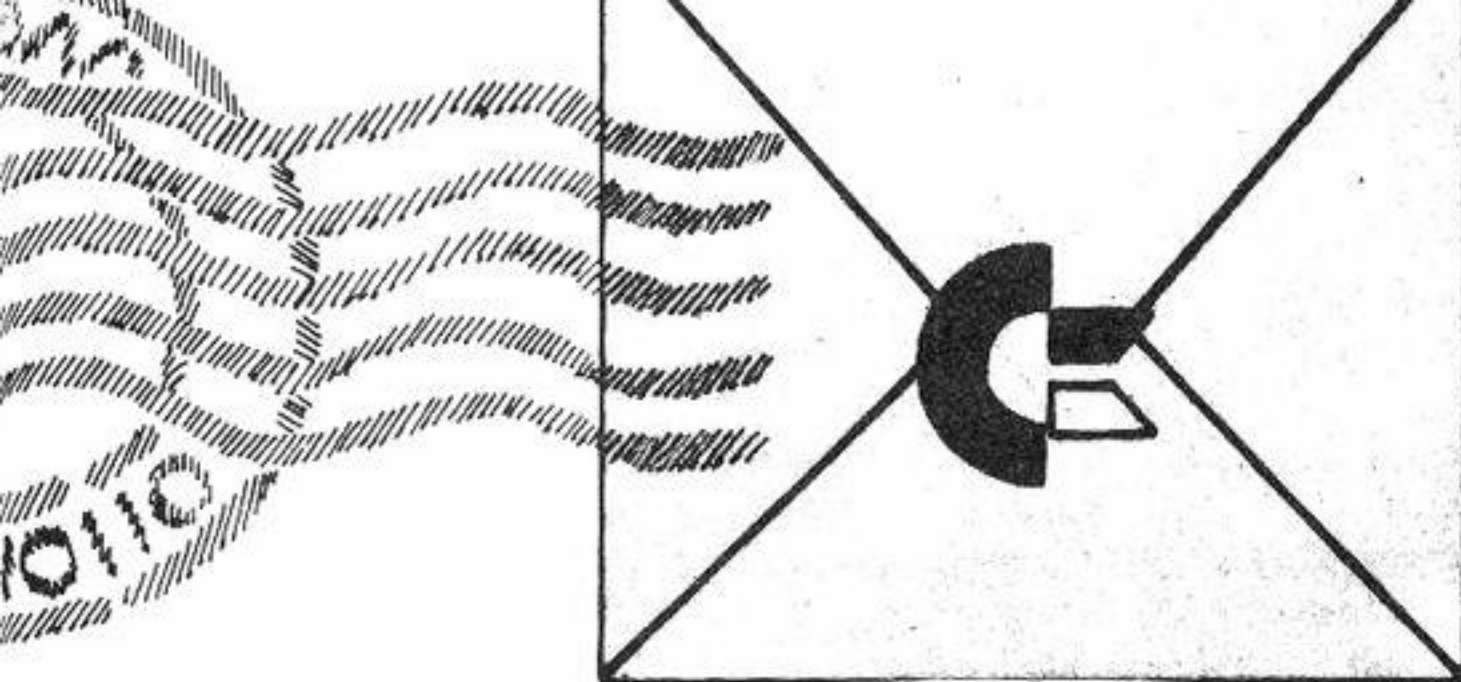
Non ci si può permettere il lusso, arrivati a questo punto di frenetiche attività collaterali, di commissionare un programma speciale, dotato di opzione crittografica, semplice e versatile. Bisogna, per forza di cose, utilizzare lo stesso programma che, di solito, si usa normalmente. Oppure ricorrere ad un programma diverso, ma pur sempre in "chiaro", in grado di essere attivato da chiunque sappia come caricare i data base, gli spreadsheet ed i word processor normalmente disponibili sul mercato.

La crittografia, quindi, pure se realizzabile con la massima facilità, sembrerebbe destinata a restare nei casi di spionaggio (poche persone e poche informazioni, ma di vitale importanza), a causa della difficoltà concreta di renderla disponibile per chi debba utilizzarla, costantemente, su vasta scala.

Il KGB e la CIA devono solo limitarsi a proteggere il "codice", e non il documento oggetto di spionaggio o controspionaggio, purchè siano pochissime le persone che lo conoscono. Avrebbero un bel da fare, invece, se volessero rendere snella una struttura accessibile a molti addetti, tra i quali la probablità di una fuga di notizie cresce in proporzione.

In casi come questi non ci troveremmo, pertanto, di fronte ad un geniale esperto che ha rintracciato un fantomatico "codice" per decodificare documenti compromettenti.

Preferiremmo pensare, infatti, che il codice glielo ha detto l'uccellino.

Alessandro de Simone

la vostra posta

Condensatore mancante

• Alcuni lettori hanno fatto notare che nell'elenco dei componenti per la costruzione dell'interfaccia RS-232 (C.C.C. n.49) è indicato un condensatore da 0.1 microfarad (C5) che, però, non è riportato nel disegno.

Precisiamo che tale condensatore, in un primo prototipo, era stato inserito, ad abundantiam, tra la massa ed il pin di alimentazione del chip (5 Volt). Nel progetto definitivo è stato invece eliminato, in quanto non risultava indispensabile. Coloro che notassero "disturbi" di vario tipo possono, ad ogni buon conto, collegare detto condensatore in parallelo alla linea di alimentazione (cioè un capo alla massa e l'altro al pin +5 volt).

Imprecisioni

□ Ho notato uno stranissimo fenomeno di natura matematica, che voglio sottoporvi per capirne il motivo...
(Uno che vorrebbe capire)

• "Effettuando la somma tra i valori 67.89 e 567.456 (assegnati, rispettivamente, alle variabili A e B) ottengo, come risultato, non il valore 635.346 -continua il nostro lettore- ma l'errato valore 635.346001.

Se, poi, impongo...
C = 635.346001
...alla richiesta Print A+B-C non ottengo il valore nullo, ma un misterioso...
-9.53674317 E-07."

Il motivo dell'evidente errore è da attribuirsi alle approssimazioni interne del computer.

Come è noto, infatti, un calcolatore lavora con la notazione binaria mentre noi siamo abituati a quella decimale. Nella "conversione" da un sistema all'altro si manifestano, per motivi che non sto qui a descrivere, inevitabili imprecisoni che vengono risolte, dai progettisti, con le approssimazioni.

Naturalmente il grado di precisone richiesto da un algoritmo in linguaggio macchina è direttamente proporzionale allo spazio occupato in memoria e al tempo necessario per eseguire la stessa routine di approssimazione.

Ne consegue che i progettisti evitano di introdurre approssimazioni molto spinte, poco frequenti in calcolatori della fascia hobbystica, e si limitano ad inserire, nelle Rom, solo le routine che ritengono sufficientemente accurate.

Capita, tuttavia, che qualche utente cerchi di elaborare calcoli apparentemente banali ma che, in realtà, non sono stati previsti dalla routine di approssimazione disponibile sul computer.

La somma dei due numeri, pertanto, non solo viene eseguita con una certa approssimazione (la differenza dal valore reale è, in effetti, modesta) ma addirittura memorizzata in una forma diversa da come viene visualizzata! Il valore che ne risulta, chiedendo la differenza A+B-D rappresenta, appunto, l'approssimazione introdotta dal sistema di calcolo, espressa in forma esponenziale.

Un home computer, pertanto, non

Ancora sul 1571

□ A differenza di quanto avete asserito, il 1571, collegato con un C/64, può formattare entrambe le facce di un dischetto. Vorrei, inoltre, precisare alcune notizie sulle porte joy.
(Luigi Russo)

• In effeti impartendo il comando...

Open 15,8,15,"U0>M1": Close 15

...e formattando, subito dopo, un disco, questo viene formattato su ambedue le facce.

Tuttavia c'è da sottolineare che per leggere lo stesso disco, in seguito, sarà sempre necessario impartire il comando (Open... eccetera) prima di qualsiasi operazione di lettura (o scrittura) sul disco stesso.

In caso contrario il computer riterrà di avere a che fare con un disco a singola faccia, con le conseguenze che puoi immaginare.

Per ciò che riguarda i connettori dei joystick, alla porta 1 corrisponde, nel C/64, la locazione 56321 ed alla porta 2 la 56320. Spostando a sinistra, inoltre, con il joy in porta 1, la leva dell'accessorio, si ottiene "anche" lo stesso effetto della pressione del tasto Control; quest'ultimo particolare consiglia di staccare il joystick durante l'uso di programmi che richiedano la pressione di detto tasto onde evitare "interferenze" accidentali.

può esser utilizzato per calcoli che richiedano una precisione rigorosa (tra cui il software di contabilità) nè per elaborazioni che interessino cifre molto piccole (chimica) o molto grandi (astronomia). Per questi scopi, purtroppo, bisogna spendere diversi milioncini per procurarsi cacolatori più potenti, precisi e veloci.

1000 icone per Amiga

Numerosi lettori chiedono che fine abbia fatto l'iniziativa "1000 icone per Amiga" con la quale si proponeva ai vari possessori di questo computer la realizzazione di icone personalizzate da raccogliere in un disco.

Molti sono i dischetti pervenuti in redazione (tra cui quelli di Giancarlo Castello di Ventimiglia, Egidio Napolitano di Taranto, Paolo Manardelli di Ravenna) e stiamo selezionando le migliori icone.

Ricordiamo di inviare icone disegnate avendo in memoria i colori standard di Amiga, cioè quelli che si ottengono per defalut inserendo il Workbench originale; in caso contrario, infatti, sorgono non pochi problemi nel memorizzare, su uno stesso disco, icone realizzate con intensità diverse di colori primari. Evitate, in ogni caso, di inviare dischetti dotati di autostart: se volete aggiungere messaggi, limitatevi a registrare semplici file di Notepad.

Per maggiori informazioni:
tel. 02/84.67.34.8

Errata Corrige

Il fascicolo "Commodore N.17" (speciale Grafica e Linguaggio Macchina) contiene alcuni errori di impaginazione, di cui abbiamo già parlato in precedenza, ma che qui ricordiamo nuovamente per coloro che si fossero procurati da poco il fascicolo in oggetto:

La pagina successiva alla n.15 è la n.17 (e non la 16).
La pagina successiva alla n.64 è la n.16 (e non la 65).
La pagina successiva alla n.23 è la n.25 (e non la 24).
La pagina successiva alla n.24 è la n.26 (e non la 25).

La tabella cui si accenna nel fascicolo, infine, è quella pubblicata su C.C.C. n. 40 (marzo '87).

Nidificazioni

☐ Non riesco ad inserire, in miei programmi Basic, un elevato numero di cicli For...Next l'uno dentro l'altro. Come posso risolvere il problema?
(Francesco Salerno - Niscemi)

• Quando si "apre" un ciclo For...Next il computer è inevitabilmente costretto a memorizzare da qualche parte, nella memoria, il nome della variabile interessata ed il suo valore, da aggiornare automaticamente dopo ogni Next.

E' quindi intuitivo che, per ragioni di praticità, occupazione di memoria e velocità operativa, i progettisti dei piccoli computer decidano di limitare il numero massimo di cicli For...Next nidificabili, vale a dire inseriti l'uno all'interno dell'altro.

Il problema, però, è facilmente risolvibile ricorrendo al semplice If...Then.

Se, ad esempio, si vogliono stampare dieci numeri, si può ricorrere al For...Next...

```
100 For i=1 to 10
110 Print i: Next
```

...oppure ad un ciclo (leggermente più lungo) If...Then:

```
100 i=1
110 Print i:i=i+1
120 If i=11 then 140
130 Go to 110
140 End
```

In questo modo è possibile simulare un ciclo praticamente infinito di cicli For...Next.

Che delusione!

☐ Ho acquistato un computer PC I che da molti punti di vista rappresenta una valida alternativa al C/64 ma che, purtroppo, presenta numerosi limiti che mi hanno molto deluso in quanto non comunicati dalle riviste del settore...
(Roberto Sabbacini - Fasagna)

• Come si può notare, noi non pub-

blichiamo solo lettere di utenti soddisfatti e felici come pasque, ma anche di coloro che, a torto o a ragione, ritengono di non essere soddisfatti dei prodotti Commodore.

Purtroppo non ho avuto modo di utilizzare il computer citato, altrimenti, lo assicuro, avrei messo in evidenza i pregi ed i difetti.

Tuttavia approfitto dell'occasione per ricordare che, progresso tecnologico permettendo, non è ipotizzabile un prodotto che, costando la metà del modello più grande (cioè il PC II), abbia le stesse caratteristiche.

Il maggior costo del PC II (o di un altro elaboratore Ms-Dos di simile potenza) e, magari, di un computer basato su un microprocessore 80286 (per non parlare di quelli ba-

Modifiche

□ Il programma l.m. pubblicato sul n.37 (che cancella lo schermo del C/64 in modo insolito) può essere simulato con il programma Basic che allego.
(Carmelo Ruggieri - Ragusa)

• Il listato (che gira solo sul C/64) è il seguente:

```
100 for i=1 to 999:print"a";:next
105 for x1=0 to 11
110 rem n=0:n=n+x1:poke781,n:sys 59903
115 n=11:n=n+x1:poke781,n:sys 59903
120 rem m=23:m=m-x1:poke781,m:sys 59903
125 m=12:m=m-x1:poke781,m:sys 59903
130 for x2=1 to 100:next:next
```

Togliendo il Rem dalle righe 110 e 120 (ed eliminando le righe 115 e 125) si ottiene l'effetto opposto.

sati sull'80386) è certamente giustificato: non mi riferisco, ovviamente, alla qualità ed affidabilità (che, in teoria, dovrebbero essere dignitose in ogni caso) ma, soprattutto, alla espandibilità (quasi nulla nel PC I), alla disponibilità di accessori vari (puramente ipotetica nei "mini" Ms-Dos), alla compatibilità con programmi che richiedono più drive contemporaneamente (o, addirittura, il disco rigido) e così via.

Il nostro lettore, comunque, stia pure tranquillo: tra breve verranno diffusi package specifici per questi piccoli Ms-Dos compatibili, che dovrebbero garantire la piena, pur se tardiva, soddisfazione degli utenti.

Per il futuro, comunque, mi sia concesso un piccolo suggerimento: leggete attentamente le nostre recensioni; dei prodotti che proviamo, di solito, riportiamo sempre le nostre impressioni, indipendentemente da presunti "ordini" ricevuti dall'alto (come sarcasticamente ipotizza il nostro deluso lettore)...

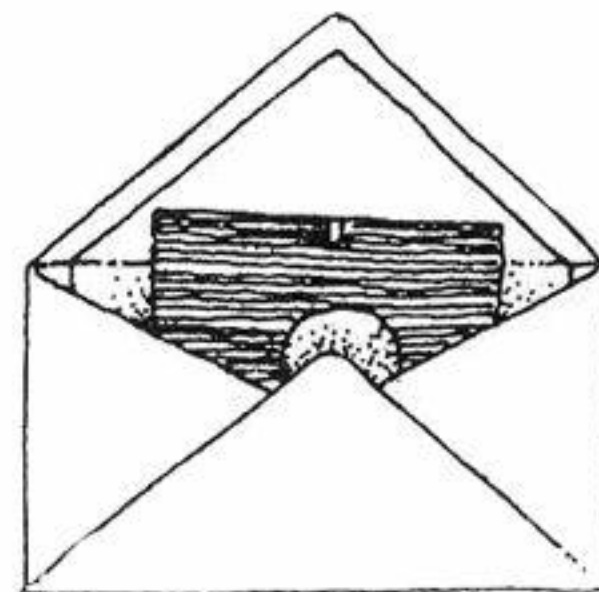
Commodore in italiano

• Ai numerosi lettori che chiedono notizie sulla disponibilità dei manuali Commodore in lingua italiana, consigliamo di rivolgersi ai "Commodore Point" (di cui è pubblicato l'elenco nelle pagine in fondo alla rivista) oppure di telefonare alla Commodore Italiana per chiedere maggiori informazioni (tel.02/61.83.21)

Magic Video

□ **Il programma "Magic Video" presente sul dischetto Zagor è davvero simpatico e mi piacerebbe utilizzarlo come presentazione in miei programmi Basic.**
(Pasquale Volpe - Agropoli e numerosi altri lettori)

• Il programma di cui si parla ha riscosso un successo inaspettato.

Purtroppo, per come è strutturato, non è possibile utilizzarlo in propri listati Basic, se non inserendo (come ultima istruzione), il comando...

Load "nome", 8

In questo modo, infatti, il programma "nome" viene caricato e mandato automaticamente in esecuzione.

Stiamo provvedendo, comunque, per fare in modo che i vari file generati dall'utente possano esser caricati e lanciati anche all'inizio di una elaborazione Basic.

Basic

Illegal quantity

□ **Aiuto! Da una rivista specializzata ho digitato il microlistato (che allego), ma il C/64 si blocca sempre con il messaggio "Illegal quantity error". Quale ne è la causa?**
(David Moroni - Faetano)

• Il programma inviato è il seguente:

```
10 Poke 646, Rnd(0)*16
20 Print Tab(Sin(x)*16+10) Chr$(209)
30 X=X+0.3: Goto 10
```

La quantità illegale è dovuta al fatto che, quando la variabile X raggiunge il valore 3.9 il valore di Sin(X) diventa negativo. La funzione Tab, invece, richiede solo argomenti non negativi.

Per far girare correttamente il programma è necessario, quindi, fare in modo che il valore racchiuso tra parentesi sia sempre compreso tra zero e 255. L'errore risiede, pertanto, nel valore 10 che deve esser modificato in 16:

```
20 Print Tab(Sin(x)*16+16) Chr$(209)
```

Non sarebbe meglio, comunque, pensarci due volte prima di definire "specializzata" una pubblicazione?...

Incontriamoci a colori

□ **Posseggo una stampante a colori Commodore MCS 801 (praticamente identica alla Seikosha GP-700A) e mi piacerebbe entrare in contatto con altri utenti che la posseggono allo scopo di scambiar programmi per utilizzarla al meglio.**
(Maurizio D'Adamo
Via di Stufa Pigna Castello, 12
03013 Ferentino -FR-)

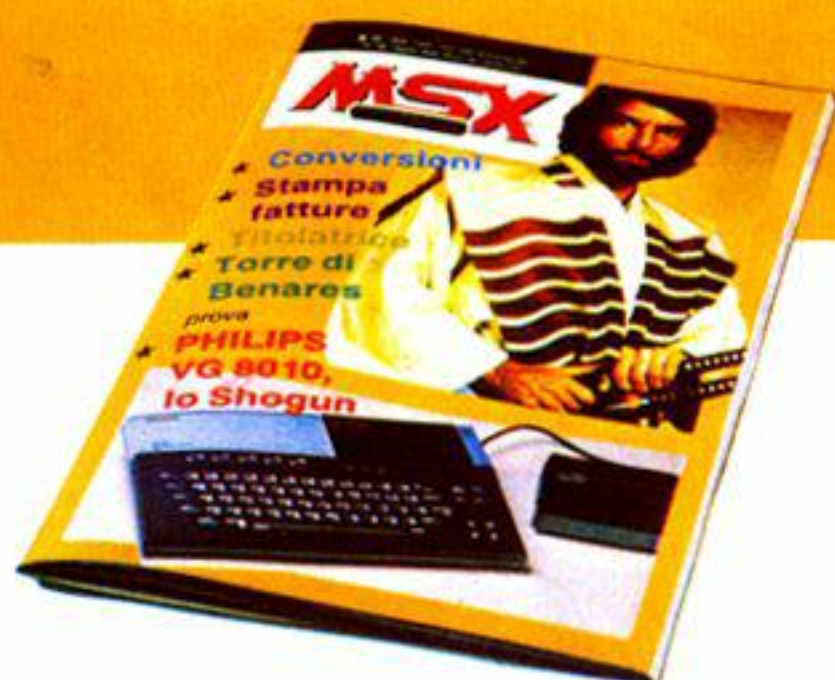
• Purtroppo le stampanti a colori non hanno avuto la diffusione sperata sia per l'elevato prezzo del nastro a colori, quasi sempre utilizzabile una sola volta, sia per la lentezza operativa.

Anche all'estero è raro trovare articoli, su riviste specializzate, che trattino i problemi connessi con le stampanti a colori.

Si può tentare di acquistare le riviste tedesche "RUN" e "64-er" oppure le americane "Compute!" e "RUN"; ma dubito che sia possibile effettuare una ricerca fruttuosa in questo campo.

128 KBYTES

SINCLAIR COM

COMMODORE

MSX

DI RIVISTA.

PUTER

Personal computer

STUDIO DI FUNZIONE

RILOCATORE DI PROGRAMMI

FUNZIONE VAL PER IL QL

TRE RIVISTE IN UNA!

E' IN EDICOLA

Systems

Personal computer è la rivista Systems per gli utenti Commodore, MSX, Sinclair. Non solo tre riviste per tre diversi utenti: **Personal Computer** è anche un'idea nuova per far comunicare tutti gli hobbisti. **Personal Computer:** 128 Kbytes di rivista, tutti i mesi in edicola. L'abbonamento cumulativo a **Computer** e **Personal Computer** costa solo L. 65.000.

Il mercato si evolve. Anche noi.

H-copy con Toma

□ Ho digitato (dal N.14) le routine grafiche di Toma che funzionano perfettamente. Ho anche digitato il programma di Hard copy della pagina grafica (C.C.C. n.48) ma il risultato che ottengo è un ammasso indistinto di punti.
(Adamo Tresca - Musile)

• La routine l.m. rilocabile pubblicata sul n. 48 suggeriva, come esempio applicativo, la copia della pagina grafica situata a partire dalla locazione 8192.

La pagina grafica delle routine di Toma, invece, è posizionata "sotto" le Rom da 57344 a 65343. E' pertanto necessario disabilitare la Rom per leggere ciò che è presente sulla Ram.

La routine ideale è quindi quella pubblicata sul fascicolo "Commodore spciale" n.17 interamente dedicato, appunto, al linguaggio macchina ed alle routine grafiche di Toma.

Soluzioni

□ Conosco alcune soluzioni dell'interessante programma "Coabitazione forzata di otto regine" (C.C.C. n. 49)
(Egidio Pignalosa - Ponticelli)

• Ecco le soluzioni inviate:

4a, 6b, 1c, 5d, 2e, 8f, 3g, 7h
6a, 8b, 2c, 4d, 1e, 7f, 5g, 3h
2a, 6b, 1c, 7d, 4e, 8f, 3g, 5h
1a, 5b, 8c, 6d, 3e, 7f, 2g, 4h
8a, 2b, 4c, 1d, 7e, 5f, 3g, 6h
5a, 7b, 2c, 6d, 3e, 1f, 4g, 8h
3a, 6b, 4c, 2d, 8e, 5f, 7g, 1h

I lettori che volessero impegnarsi per scoprirne altre, potranno inviarle in redazione.

Sys inventate

□ Ritengo di aver inventato una Sys particolare (Sys 45876,67) che "blocca" il C/64 costringendo a spegnerlo e riaccenderlo. Che cosa provoca questa Sys?
(Enrico Ferri - Brindisi)

• Quando si impartisce una Sys, il computer "abbandona" il Basic per lavorare in linguaggio macchina.

Ciò significa che, da quel momento, elabora i byte presenti a partire dalla locazione indicata, considerandoli come una successione di codici macchina.

Poichè una locazione di memoria contiene sempre un valore, magari casuale, il microprocessore inizia l'elaborazione con conseguenze altrettanto "casuali" una delle quali (che è la più frequente) è proprio il blocco del sistema che costringe al reset generale.

Digitando a casaccio il comando Sys, pertanto, è molto facile incappare in inconvenienti del genere lamentato.

Due dita o dieci?

□ Non capisco perchè, dal momento che Lui ci ha dato dieci dita, i progettisti di computer si ostinano ad usare il sistema binario (basato su due sole... dita).
(Leonardo Losito - Gioia del Colle)

• Quando si pensò di realizzare un sistema logico supportato dall'elettronica, la tecnologia di quel tempo offriva i relè e le valvole termoioniche.

Questi dispositivi potevano garantire, con una certa sicurezza, il solo riconoscimento di un passaggio di corrente elettrica (cioè lo stato logico 1) oppure la sua assenza (stato 0). L'unica struttura logico-matematica che si poteva applicare, quindi, risultò quella binaria che, anzi, fu riscoperta per l'occasione dal momento che nel passato, ovviamente(!), era stata studiata più che altro come una semplice curiosità matematica.

Naturalmente molti tecnici, matematici e scienziati si buttarono a capofitto per approfondire alcuni argomenti che, in seguito, furono applicati agli elaboratori.

Si verificò, quindi, il solito fenomeno del serpente che si mangia la coda: la "teoria" spingeva i costruttori a proseguire in applicazioni binarie, e le nuove macchine offrivano il terreno giusto per ampliare la ricerca teorica che si specializzava sempre più nel calcolo binario.

Allo stato attuale della tecnologia elettronica, però, sarebbe possibile costruire dispositivi elettronici in grado di "riconoscere" non solo l'assenza o il passaggio di corrente, ma anche più "stati" elettrici intermedi. Al giorno d'oggi, insomma, non sarebbe un problema realizzare dispositivi "decimali".

Tuttavia sarebbero necessari investimenti consistenti di svariati milioni di dollari che potrebbero essere recuperati, commercialmente, solo dopo un periodo piuttosto lungo: troppo lungo, comunque, per distogliere gli stessi milioni di dollari da ricerche che permettano recuperi più immediati.

Per una questione (come al solito)

Consigli per C/128

Basic

☐ **Una mia piccola scoperta potrebbe essere utile per i possessori di C/128...**
(Pierosoft Firenze)

• Per trascrivere i listati contenenti numerose istruzioni di tipo Data - prosegue Piero- digitate, in modo diretto...

Key 1,"Data": Key 3,",": Auto 10

...in modo che, con il C/128, sarà possibile servirsi della sola zona della tastiera riservata al tastierino numerico; la velocità di battitura risulterà decisamente maggiore.

I consigli sono validi anche per C/16 e Plus/4 che, tuttavia, non possedendo il tastierino, non potranno offrire la comodità offerta dal C/128.

di soldi, saremo quindi costretti, almeno per alcuni decenni, a ragionare come il computer: in binario!

Tacchetta di protezione

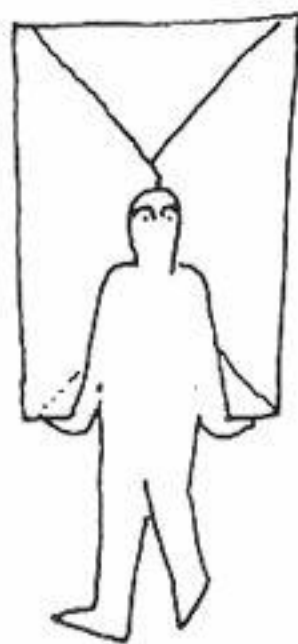
☐ **Come può, un computer, riconoscere che la tacchetta di protezione di un dischetto è applicata oppure no?**
(Leda Litucci - Foggia)

• In un drive, in corrispondenza della tacchetta, è presente un deviatore che, a seconda del modello, è elettromeccanico, ottico, oppure magnetico.

Quello elettromeccanico non è altro che una semplice linguetta che, se trova la fessura, si abbassa e se, invece, è presente la tacchetta, rimane sollevata; alla linguetta è collegato un deviatore elettrico che, collegato opportunamente al sistema operativo del drive, consente di conoscere lo "stato" del dischetto.

In modo analogo agisce il metodo ottico che presenta una cellula fotosensibile; questa, di solito illuminata da un led, non riceve più la luce se la fessura del floppy è "oscurata" dalla tacchetta.

I drive dei sistemi AT compatibili, poi, hanno come sensore un apparato magnetico per cui l'applicazione di una semplice fascetta di carta non è sufficiente: è infatti necessario che la tacchetta sia di carta metallizzata, altrimenti il drive non si "accorge" della sua presenza.

E' quindi possibile formattare un disco anche se è presente la tacchetta di protezione (di qualunque tipo); è sufficiente individuare il sensore del drive e manometterlo opportunamente affinchè lo stesso drive "ritenga" di operare con un floppy idoneo per la registrazione.

Cartuccia troppo grossa

☐ **Vorrei acquistare la cartuccia sprotettrice Freeze Frame ma, a causa della dimensione della fessura del mio C/128, non è possibile inserirla. Come posso fare?**
(Gianni Ronchi - Senago)

• Purtroppo alcune cartucce sono dotate di una custodia voluminosa e presentano il problema lamentato.

In questi casi, tuttavia, dovrebbe esser possibile svitare la custodia stessa ed estrarne il circuito stampato, di solito piuttosto sottile. Naturalmente è necessario prestare la massima attenzione nell'eseguire l'operazione, soprattutto quando è necessario rimuovere la scheda dal retro del computer.

Gw-Basic non funzionante

☐ **Ho acquistato il vostro Ms-Dos emulator ma, non riuscendo a caricare il Gw-Basic, ho esaminato, con un apposito Disk-Tool, il dischetto, che presenta numerosi errori.**
(Filippo Oliva - Catania)

• I dischetti posti in commercio sono prodotti industrialmente, e non uno per uno.

Può quindi verificarsi il caso di alcune copie malfunzionanti che, comunque, noi sostituiamo immediatamente (e gratuitamente) non appena la copia difettosa, completa in ogni sua parte, viene inviata in redazione.

Ho notato, tuttavia, che molti edicolanti espongono al sole anche le pubblicazioni contenenti supporti magnetici (nastri e dischi); con il sopraggiungere della bella stagione la temperatura raggiunge valori elevati, che possono smagnetizzare alcune zone del supporto.

Gli edicolanti dovrebbero tenere al riparo dal sole simili pubblicazioni, ma non sempre ciò si verifica.

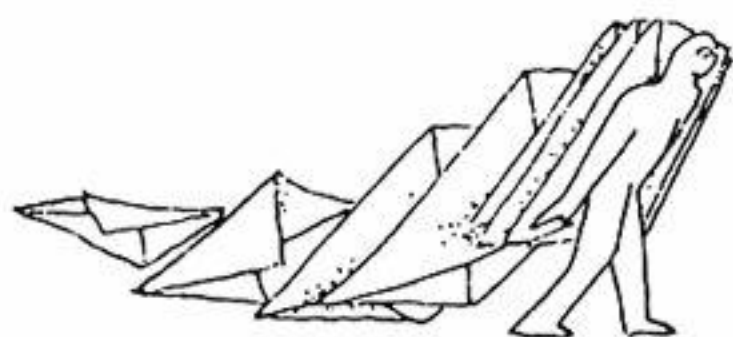
Errori

☐ **Non riesco a digitare i vostri listati. Forse contengono protezioni?**
(Pitro Parisi - Messina)

• Assolutamente no, diamine! Sarebbe una cattiveria disumana costringere i nostri lettori a digitare listati che poi non girano.

Dalla lettera, tuttavia, ritengo che il nostro lettore sia alle prime armi, tant'è vero che non indica nemmeno il tipo di errore segnalato dal computer, impedendo, di fatto, un qualsiasi mio intervento.

Non posso che ripetere i soliti consigli:

• Visualizzare il listato in successivi blocchi di cinque righe ciascuno.
• Verificare accuratamente che ogni riga sia perfettamente identica a quella pubblicata sulla rivista.
• Effettuare le correzioni del caso e premere il tasto Return dopo ogni correzione.
• Visualizzare nuovamente la riga corretta per verificare che, effettivamente, il computer abbia accettato

la correzione.
• In caso di estrema necessità, inviare su supporto magnetico il programma digitato, in modo da permettermi di esaminare gli errori.

Un micro da 16 bit

□ E' possibile, utilizzando la porta giochi del C/64, inserire una scheda dotata di processore a 16 bit?
(Dino Calvisi - Petogna)

• In teoria la risposta dovrebbe essere affermativa, ma la spesa da sostenere sarebbe, complessivamente, ben superiore alla cifra da spendere per entrare in possesso di un "vero" computer basato su un microprocessore più potente, magari dotato di drive, monitor e stampante!

HACKER, l'ultima sfida

Sul dischetto "Commodore 64 Club" #5, il prodotto bimestrale per utenti di C/64-128, questo mese è presente, oltre a diversi videogiochi ed utility, anche un programma un po' particolare; si tratta di un software che metterà a dura prova la vostra capacità di "Hacker".

Per gli "smanettoni" appassionati di Informatica, il gioco preferito è molto spesso lo sproteggere il maggior numero di programmi possibile.

L'intenzione che anima gli "Hackers", ovvero gli sprotettori, è quella della sfida, di un costante confronto con se stessi e con le proprie capacità; le ricompense sono poi date dal vedere il loro nome di battaglia nelle videate di presentazione e dall'aver "personalizzato" questo o quel programma.

Questo software, Hacker appunto, vuole dare a tutti coloro che hanno l'hobby della "sprotezione" un'occasione per dimostrare tutta la loro bravura.

Non si tratta quindi di un vero e proprio videogame ma di una "sfida" alla quale invitiamo tutti gli affezionati utenti dei nostri prodotti su disco.

Il programma, non appena partito richiede una prima password (parola chiave) che, una volta indovinata, vi permetterà di ascoltare una gradevole musica classica.

La seconda fase del programma richiederà due parole chiave e, al termine di questa fase, verrà visualizzata una schermata con diverse scritte "Systems".

Per considerare compiuta la missione, l'aspirante sprotettore deve scoprire le tre parole chiave, sostituire la musica classica con quella presente sul disco con il nome "MUSIC" e sostituire la videata "Systems" con una in cui compaia il suo nome di battaglia.

I primi cento che invieranno un dischetto con il programma sprotetto e un allegato contenente le parole chiave, avranno una simpatica sorpresa ma soprattutto avranno l'opportunità di veder pubblicato il loro nome nelle prime pagine della rivista Commodore Computer Club.

Accetti la sfida o pensi che per te sia troppo difficile?

Il materiale va inviato a:

SYSTEMS EDITORIALE
Rubrica "Hacker"
V.le Famagosta, 75
20142 MILANO

Il programma per pagare le tasse

• Ai numerosi lettori che chiedono informazioni in merito, ricordiamo che sarà presto in edicola il programma 740 (per C/64) riveduto ed aggiornato secondo le disposizioni di quest'anno.

Non sarà possibile fornire le modifiche da apportare alla versione precedente perchè, anno per anno, il programma è sottoposto a profonde modifiche che costringono a sostanziali rifacimenti di intere sezioni del listato.

risposte rapide

Istruzioni in italiano
(Angelo Di Nieri - Villalba)

L'indirizzo della Commodore Italiana, cui richiedere informazioni sulla disponibilità di libretti di istruzioni in italiano, è:
Via Fratelli Gracchi n.48
20092 Cinisello Balsamo (Mi)
Tel. 02/61.83.21

Alfabeti stranieri
(Edmondo Bedocchi - R.Emilia)

Dubito che sia possibile trovare facilmente, per il C/64, stampanti che possano riprodurre simboli matematici e caratteri alfabetici inconsueti.

Stampante Laser TP-80
(Edoardo Tusa - Catania)

Non conosco la stampante citata e, di conseguenza, non posso essere di alcun aiuto. Per il futuro consiglio di leggere attentamente le nostre impressioni sulle apparecchiature provate, in modo da far tesoro dei nostri consigli (chiedo scusa per l'immodestia).

Computer Laser 500 M
(Nicola Piras - Nulvi)

Questa rivista può trattare esclusivamente argomenti inerenti i computer Commodore; ne consegue che non ci sarà mai possibile affrontare il computer Laser che possiedi.

Directory per Plus/4
(Simona Bitocchi - Tivoli)

Il nostro dischetto "Directory" contiene, mese per mese, tutti i programmi pubblicati sulla rivista Commodore Computer Club con la sola eccezione dei programmi scritti per Amiga (per ovvi motivi di incompatibilità). I programmi "universali" e quelli spcifici per C/16 potranno quindi girare senza problemi sul tuo Plus/4.

Fare a meno della carta
(Tommaso Pagliani - S.Vito)

Non penso che sia possibile, con un semplice C/128, pur se dotato di drive, fare a meno della carta e memorizzare diagrammi, schemi e disegni solo su supporto magnetico. A parte la velocità operativa di ricerca dei dati, il computer non è sufficientemente versatile per fare ciò che chiedi: memorizzare schermate grafiche su un dichetto risulta forse più costoso delle fotocopie che è possibile effettuare (senza considerare il tempo richiesto per realizzare i lunghi "trasferimenti").

Informazioni scarse
(Andrea Aloisi - Avezzano)
(Marco Gigante - Cavallino)

I "sintomi" descritti sono pochi e generici: non mi è possibile risalire alla causa dei malfunzionamenti lamentati.

Controllo errori
(Gianni Ronchi - Senago)

I programmi contenenti una lunga serie di Data sono destinati, per lo più, ad esperti che sono in grado di controllare la validità della trascrizione servendosi anche del disassemblato commentato che, come puoi notare, accompagna la quasi totalità dei listati l.m. pubblicati sulla rivista.

Syntax Error in Toma
(Salvatore Mariano - Paola)

Prima di digitare un qualsiasi programma che utilizzi i comandi grafici delle routine di Toma, è indispensabile caricare (e lanciare) le routine in linguaggio macchina, pena la comparsa del messaggio "Syntax error".

Compiti difficili
(Antonio A. - Mandello)

Per digitalizzare l'immagine proveniente da un ecografo è necessario conoscere le caratteristiche del segnale di uscita dell'ecografo stesso e, naturalmente, accontentarsi della risoluzione offerta dal C/64, a patto che il miracolo del trasferimento si verifichi.

Solo per drive
(Giorgio Scoppitti - Roma)

I programmi in l.m. che pubblichiamo, tra cui quelli che utilizzano tecniche di Autorun, richiamano diverse routine specifiche del drive ed occupano, a volte, la zona Ram riservata alla gestione del registratore; ne consegue che non è quasi mai sufficiente cambiare il numero di device (da 8 a 1) per vederli funzionare con il datassette.

Registrazione aliena
(Roberto Francini - Prato)

La gestione del registratore è totalmente diversa per i computer C/128 e C/16. Ne consegue che il trasferimento di dati e programmi è possibile solo tramite drive; a meno che qualche lettore di buona volontà non decida di inviare una routine di simulazione...

In compagnia di Amiga

Grosse novità sono previste in arrivo entro il prossimo autunno; quasi tutti i prodotti riguardano l'Amiga.

Un disco rigido, naturalmente esterno, potrà affiancarsi all'Amiga 500; si tratta del modello A-590, da 20 megabyte.

Dopo il successo dell'adattatore telematico per il C/64 (e dopo reiterate proteste dei neo-possessori di Amiga) vedrà la luce uno specifico adattatore per A-500 e, forse, per A-2000; l'apparecchio sarà omologato SIP e dotato di software pronto per effettuare rapidi collegamenti telefonici.

Nel campo professionale dovrebbe arrivare, finalmente, la scheda (solo per A-2000) in grado di rendere il computer idoneo a "trattare" lo standard 286; stiamo parlando della scheda A-2286 che renderà (come Janus per IBM-XT) l'Amiga 2000 compatibile AT.

Numerosi monitor saranno presto disponibili: monocromatici, a colori, a lunga e bassa persistenza, interamente compatibili con lo standard (ed i cavetti...) C/64, Amiga, Ms-Dos; ci sarà solo l'imbarazzo della scelta.

Passando al settore dei videogame, l'accessorio che desterà più interesse sarà sicuramente un nuovo joystick (sempre per Amiga) che renderà più agevole la gestione dei giochi per i quali il mouse non risulta lo strumento di controllo più idoneo.

Un apparecchio altamente professionale (come il prezzo...) farà venire l'acquolina in bocca agli appassionati di grafica computerizzata e di videoregistrazione. Ci riferiamo al Genlock, apparecchio che consente di miscelare immagini video provenienti da più fonti diverse: telecamere, videoregistratori, computer, videodischi e così via. La categoria ideale degli utenti è rappresentata dai network televisivi, dagli studi di ripresa per filmati pubblicitari, da professionisti "creativi" di vario tipo che hanno semepre a che fare con immagini digitalizzate.

Dovrebbe anche arrivare anche una nuova versione di Workbench (la 1.3), prematuramente annunciata, ma ancora in fase di verifica.

La stampante MPS-1250 accontenterà i possessori di C/64 che hanno intenzione di "passare" all'Amiga: sul retro della macchina, infatti, sono presenti due connettori; il primo permette il collegamento con il C/64 (oppure C/128, C/16 o Plus/4), mentre il secondo garantisce la perfetta compatibilità con l'Amiga. Non sarà quindi necessario cambiare anche la stampante quando si compirà il gran passo...

Nel campo dello standard Ms-Dos, la Commodore spera di ripetere, in Itàlia, lo straordinario successo riscosso in Germania; notevole impulso è stato infatti conferito alla produzione dei computer basati sul microprocessore 80386.

Per il PC 6040 sarà disponibile il famoso package Windows 386, corredato di mouse. Il PC 40, invece, potrà esser fornito con il processore matematico per potenziarne le risorse richieste dal software specifico.

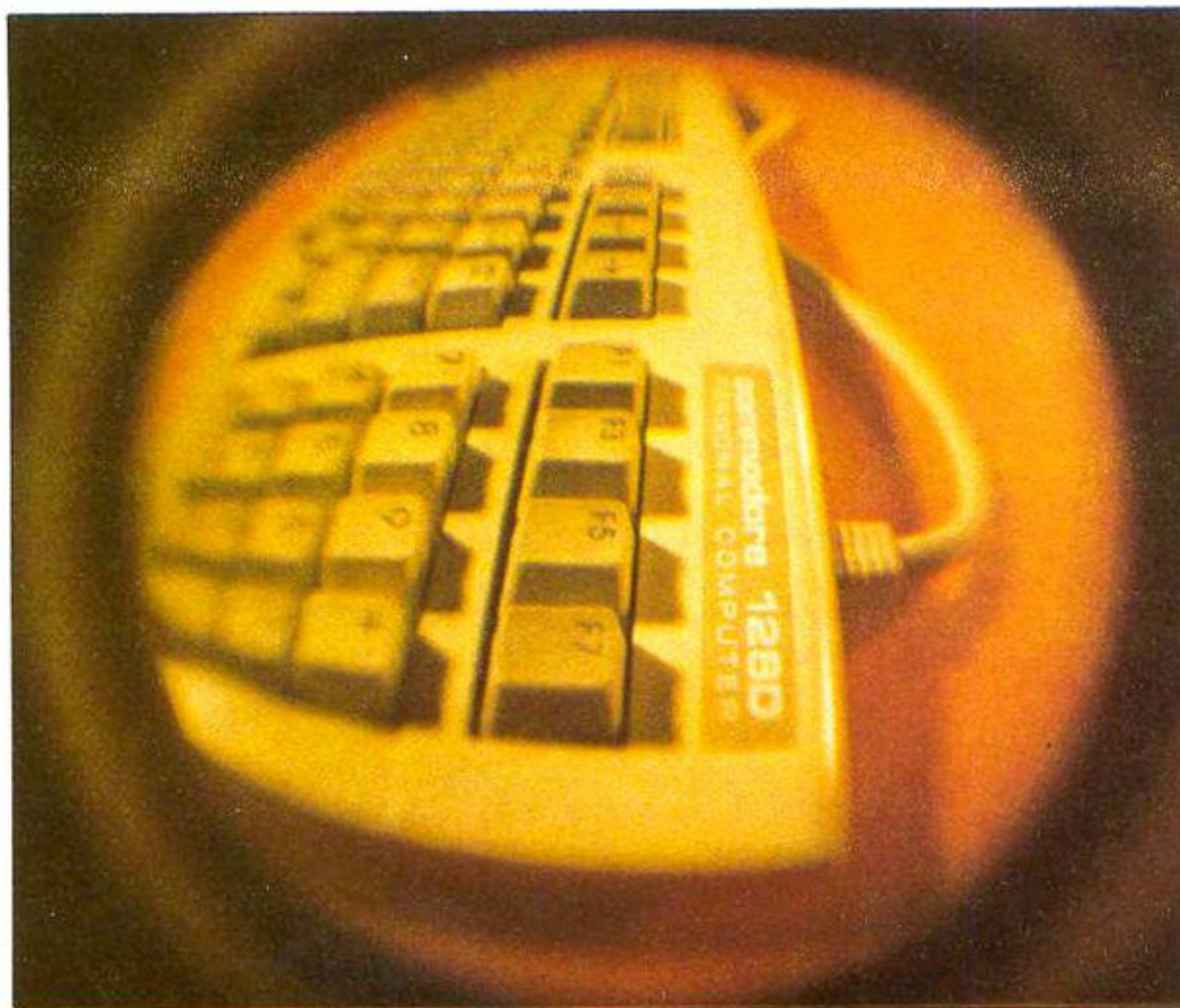
Commodore e lo Sport

Continua l'intensa attività della Commodore nel settore delle sponsorizzazioni.

Lo sport scelto non poteva che essere il motociclismo, da sempre considerato attività "giovane" ed avventurosa.

Massimo Bolfante, infatti, piloterà una Honda-Commodore 125 appositamente giunta dal Giappone per misurarsi nel difficile Trofeo Grand Prix.

Il primo appuntamento si è svolto il 13 marzo a Misano Adriatico (Fo); le sfide successive si verificheranno a Vallelunga (3 aprile); Magione (1 maggio); Monza (29 maggio); Mugello (5 giugno); di nuovo Maisano Adriatico (10 luglio e 14 agosto).

Al coraggioso pilota indirizziamo un caloroso "In bocca al lupo!", anche da parte di tutti gli aficionados Commodore.

TRE PROGRAMMI PER CHI INIZIA

Queste tre semplicissime routine si prestano per esser modificate a seconda delle esigenze

di **Claudio Lanzoni**

Simula Sprite

Il programma "Simula sprite" consente di riprodurre (a velocità, purtroppo, ridottissima) il movimento di uno sprite rotante su sfondo in alta risoluzione, senza cancellare le figure che attraversa.

Essendo il C/16, come è noto, privo di sprites, è necessario, lavorando in schermo grafico, ricorrere all'uso intensivo delle istruzione SSHAPE e GSHAPE.

L'istruzione SSHAPE permette di memorizzare, in una variabile alfanumerica, il contenuto di un'area rettangolare posta in qualsiasi punto dello schermo grafico.

La sintassi è la seguente:
SSHAPE var.stringa,X1,Y1,X2,Y2
in cui X1 e Y1 sono le coordinate del vertice opposto al vertice X2 e Y2.

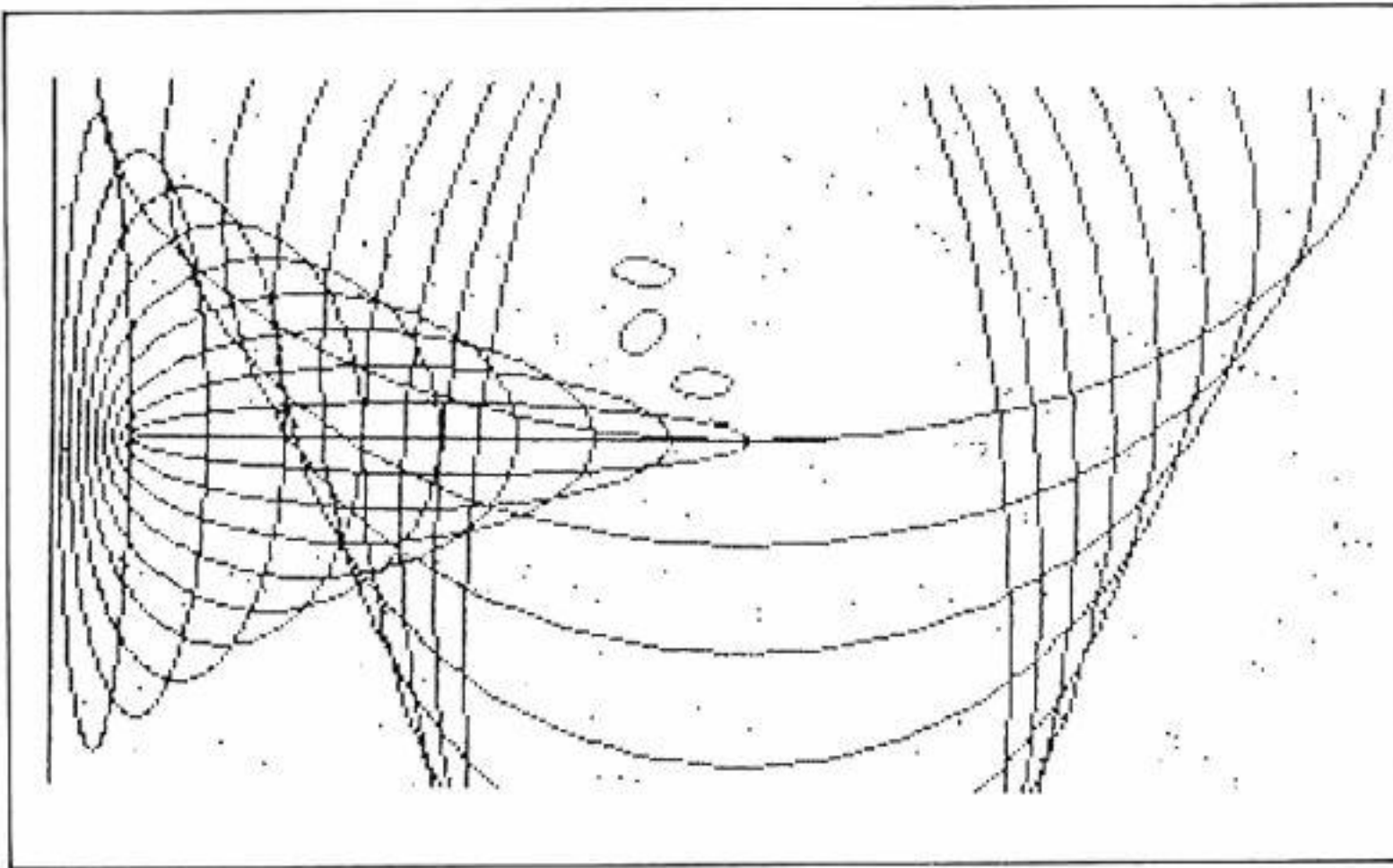
Tuttavia, siccome la lunghezza massima di una stringa Basic è limitata a 255 caratteri, l'area salvabile è limitata.

L'istruzione GSHAPE, invece, serve a visualizzare il contenuto del rettangolo prima memorizzato con l'istruzione SSHAPE.

La sintassi è la seguente:
GSHAPE stringa,X,Y,modo
in cui la stringa non è necessariamente una variabile, X e Y sono le coordinate del vertice in alto a sinistra dell'area salvata, mentre il "modo" può variare da 0 a 4, con i seguenti significati:

0 = riproduce l'area così com'è
1 = riproduce l'area in campo inverso
2 = esegue l'OR tra l'area salvata e l'area dove viene posizionata
3 = esegue l'AND tra l'area salvata e quella dove viene posizionata
4 = esegue l'EOR (exclusive OR) tra l'area e quella della sua posizione.

Il modo, se non specificato, è settato a zero.

Vi consigliamo, comunque, di consultare il manuale di istruzioni del computer per ulteriori e più detta-

```
100 REM **********************************
110 REM *    SIMULAZIONE DI SPRITES      *
120 REM * PER C/16, PLUS/4 E C/128 (!)   *
130 REM *  BY CLAUDIO LANZONI @ 1988     *
140 REM *    MASSALOMBARDA (RAVENNA)     *
150 REM **********************************
160 TRAP440:COLOR0,15,0:COLOR4,6,4:COLOR1,2,7
170 GRAPHIC1,1:SCALE0
180 REM C16: 0OKE65286,PEEK(65286)AND239
190 REM C/128: FAST
200 FORI=1TO200
210 DRAW1,INT(RND(0)*320),INT(RND(0)*200):NEXT
220 Y=100:FORI=0TO100STEP10
230 CIRCLE1,I,100,X,Y:X=X+8:Y=Y-10:NEXT
240 Y=100:X=150:FORI=0TO160STEP20
250 CIRCLE1,160,I,X,Y:X=X*.9:Y=Y*1.1:NEXT
260 X=160:Y=100
270 REM C/16: POKE65286,PEEK(65286)OR16
280 REM C/128: SLOW
290 IF X>7 AND Y>7 THENSSHAPEG$,X-7,Y-7,X+7,Y+7
300 IFX>7ANDY>7THENCIRCLE1,X,Y,7,4,,,Z,30
310 IFJOY(2)=0THEN310
320 IFJOY(2)=1THENGOSUB420:Y=Y-1:Z=Z+15
330 IFJOY(2)=5THENGOSUB420:Y=Y+1:Z=Z+15
340 IFJOY(2)=3THENGOSUB420:X=X+1:Z=Z+10
350 IFJOY(2)=7THENGOSUB420:X=X-1:Z=Z+10
360 IFJOY(2)=2THENGOSUB420:Y=Y-1:Z=Z+20:X=X+1
370 IFJOY(2)=4THENGOSUB420:Y=Y+1:Z=Z+20:X=X+1
380 IFJOY(2)=6THENGOSUB420:Y=Y+1:Z=Z+20:X=X-1
390 IFJOY(2)=8THENGOSUB420:Y=Y-1:Z=Z+20:X=X-1
400 REM  IFJOY(2)=128THEN290
410 GOTO290
420 IFX>7 AND Y>7 THEN GSHAPEG$,X-7,Y-7
430 RETURN
440 GRAPHIC0,1
450 REM C/16 :POKE65286,PEEK(65286)OR16
460 REM C/128: SLOW
470 PRINTERR$(ER)"IN"EL:HELP:END
```

gliate informazioni.

L'utilizzo del programma è semplicissimo: dato il RUN, il computer disegna un grafico.

Al termine apparirà un ovale (sprite) che potete far rotolare in lungo e in largo sullo schermo tramite joystick in porta 2; col pulsante di fuoco è possibile fissare stabilmente una posizione della figurina.

Sono accettati anche i comandi di direzione diagonale.

Vi sono alcuni controlli, per evitare che lo "sprite" vada oltre il bordo del video, che sarebbe opportuno studiare con attenzione.

Il programma gira anche sul C/128; nonostante questo computer disponga di sprite (e della relativa gestione super-veloce) la routine è tuttavia interessante per studiare procedure di riproduzioni di piccole aree di schermo.

Commento al listato

160: vengono settati i colori e, in caso di errore, si salta alla linea 220
170: si entra in modo grafico e viene azzerata la scala
200-260: si disegna un grafico ottenuto con cerchi, ovali e punti
290: viene memorizzata, nella variabile G$, la porzione di schermo entro la quale verrà stampata la nostra figurina
300: viene disegnato lo sprite
310-410: si controlla in quale direzione è stato premuto il joystick 2, e si agisce di conseguenza
420: subroutine che, prima di aggiornare la posizione della figurina, stampa la variabile G$, che contiene lo sfondo coperto dallo sprite
440-470: è la routine che viene attivata in caso di errore.

Nel programma principale sono utilizzate le variabili: PX e Y (coordinate del centro dello sprite), Z (angolo di rotazione dello sprite) G$ (contenuto dell'area salvata).

Caratteri

Il programma "Deformatore di Caratteri" prende spunto da un programma pubblicato in un vecchio libro sul C/16 e ne rappresenta lo svi-

```
B
COMMODORE COMPUTER CLUB
COMMODORE
COMPUTER
CLUB
```

```
100 REM **********************************
110 REM *    DEFORMATORE DI CARATTERI    *
120 REM * BY C.LANZONI MASSALOMBARDA (RA)*
130 REM *    PER: C/16, PLUS/4, C/128    *
140 REM **********************************
150 ZE=0:UN=1:OT=8:DU=2
160 NX=40:P=8:TRAP420:NY=25:Q=8
170 GRAPHIC1,1:XX=0:YY=1
180 GETKEYA$
190 IFA$=CHR$(13)THENXX=0:YY=YY+Q/8:GOTO180
200 IFA$=CHR$(20)THEN360
210 IFA$=CHR$(27)THEN380
220 IFA$=CHR$(19)THEN400
230 IFA$=CHR$(130)THEN170
240 GOSUB270
250 XX=XX+P:IFXX>320-PTHENXX=0:YY=YY+Q/8
260 GOTO180
270 REM C/128: FAST
280 C=500:CHAR,ZE,ZE,A$
290 FORX=ZE TO P-UN
300 FORY=ZETOQ-UN
310 LOCATEX*OT/P,Y*OT/Q
320 DRAWRDOT(DU),XX+X,YY*OT+Y
330 NEXTY:SOUNDUN,C,DU:C=C*1.05:NEXTX
340 REM C/128: SLOW
350 RETURN
360 XX=XX-P:YY=YY+(XX<0):XX=XX-320*(XX<0)
370 A$=" ":GOSUB270:GOTO180
380 GETKEYA$,B$:NX=VAL(A$)*10+VAL(B$)
390 P=319/NX:GOTO180
400 GETKEYA$,B$:NY=VAL(A$)*10+VAL(B$)
410 Q=191/NY:GOTO180
420 GRAPHIC0:REM C/128: SLOW
430 PRINTERR$(ER)"IN"EL:HELP:END
```

luppo ed il complemento; permette, infatti, di ingrandire o rimpicciolire ogni simbolo dell'alfabeto Commodore.

Come sia possibile tale "prodigio" è presto detto: la funzione basic RDOT(N) risolve da sola tutti i problemi.

I calcoli che il programma esegue sono estremamente semplici, come vedremo in seguito.

Il programma si avvale della funzione numerica RDOT(N), che "informa" sulla attuale posizione del Pixel Cursor (PC). Nel nostro caso la funzione è usata con indice 2, per determinare il colore (0 oppure 1 se si lavora in alta risoluzione; 0, 1, 2, 3 se si lavora in modo multicolore) della attuale posizione del PC.

In ordine, gli indici di RDOT(N), restituiscono:

RDOT(0) = coordinata X del PC
RDOT(1) = coordinata Y del PC
RDOT(2) = colore del PC

Il programma fa in modo che il PC attraversi, uno alla volta, i pixel della lettera presente nella cella in alto a sinistra dello schermo grafico, così da avere la possibilità di riprodurre ogni pixel della lettera nello stesso colore (0 = pixel spento, 1 = pixel acceso) un quantitativo di volte proporzionale al totale di caratteri che si desidera appaiano in una riga o in una colonna dello schermo.

Per selezionare il numero di caratteri di una riga, è necessario premere il tasto ESC e, subito dopo, due cifre indicanti, appunto, il numero dei caratteri che l'utente vuole che stiano nella riga.

Per selezionare il numero di caratteri di una colonna, occorre invece premere il tasto Home, poi inserire le due cifre riguardanti il numero di caratteri desiderati.

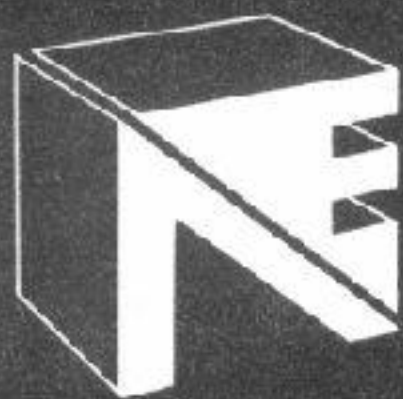
Ecco due esempi che chiariscono definitivamente l'utilizzo dei comandi:

ESC 09: "dilata" orizzontalmente i caratteri in modo che in una riga vi siano esattamente nove caratteri.

HOME 32: "restringe" verticalmente i caratteri per far sì che in una colonna vi siano 32 caratteri (il numero di caratteri standard per colonna è di 25).

Nella versione di queste pagine il programma è in grado di alterare soltanto le lettere ed i caratteri semigrafici dell'alfabeto Commodore, ma la questione è facilmente risolvibile: occorre posizionare il set dei nuovi caratteri in una zona di memoria al di fuori dallo schermo grafico (che viene utilizzato per la stampa dei mini e dei mega caratteri) e situata dopo la fine del programma Basic.

Lo spazio rimanente (per il C/16) non è molto, ma si dovrebbe riuscire ugualmente a creare il proprio ristretto e non completo set.

Per fare in modo che i caratteri

NEGOZIO AL PUBBLICO
E VENDITA PER CORRISPONDENZA
CASH & CARRY

COMMODORE AMIGA 500
AL PREZZO PIÙ BASSO D'ITALIA »CON GARANZIA & OMAGGIO«

Amiga-VID — L. 139.000
Digitalizzatore di immagini per Amiga 500/1000/2000

Amiga Syntetic — L. 170.000
Digitalizzatore Audio per Amiga 500/1000/2000

Amiga VIDEOSOUND — L. 290.000
Digitalizzatore Audio + VIDEO, tutto in uno.
Ottimo, per 500/1000

INT. MIDI AMIGA — L. 89.000
Nuova interfaccia midi per Amiga 500/1000/2000

DRIVE AGGIUNTIVO AMIGA 500/1000 (SLIM LINE)
235.000

ESPANSIONE 512 K PER AMIGA 500
INT. con orologio **L. 149.000**

KICKSTART V.I.2 + ESP. 256 K.
per **AMIGA 1000** **L. 275.000**

EMULATORE 64 per AMIGA — L. 49.000
L'UNICO EMULATORE VERAMENTE FUNZIONANTE CON L'AUDIO E CON LA POSSIBILITÀ DI SALVARE I FILES SU 3 1/2

VD-AMIGA II — **L. 950.000**
Novità digitalizzatore in tempo reale per A 500/1000/2000

NOVITÀ **PENNA OTTICA PER AMIGA 500/1000/2000 L. 149.000**

NOVITÀ **TITLE SAVER PLUS « per Amiga** — **L. 199.000**
(Orologio + testi funzione e molto più!!!

DRIVE INTERNO PER AMIGA 2000 (MECC. NEC)
L. 199.000

Cavo stampante 500/1000/2000 — **L. 25.000**
Cavo Monitor 500/1000/2000 — **L. 30.000**
Modulatore TV per Amiga — **L. 49.000**
HARD DISK per A500/1000/2000 L. TELEFONARE

COMMODORE 64/128 - COMMODORE 64/128

THE NEW FINAL TURBO III
per 64/128 (modo 64) **L. 69.000**
L'evoluzione continua!!!
Eccovi l'ultima release della mitica cartuccia notevolmente migliorata e modificata. Turbo la favolosa routine dello speddos su cartuccia fino a 10 volte più veloce sia in lettura che in scrittura!! 8 tasti funzione programmati. 24 K ram per i prog in Basic. Un favoloso protettore di programmi tipo O.M.A. incorporati. Dischi e cassette IN UN SOLO FILE!! (+ boot se necessita. Inoltre ha incorporato il GAME KILLER (evita la collisione degli sprite ed ha ben 40 comandi Basic Turbo a disposizione ...HARDCOPY "HP". Premendo un solo tasto potrete fare l'hardcopy del video in 12 gradazioni di grigio.
ECCEZIONALE!!!

PROCESSORE VOCALE (VOICESYNTETIC)
L. 115.000
Digitalizzatore vocale tipo "Voice Master" notevolmente migliorato composto a interfaccia hardware + microfono software interamente in italiano con ampio manuale di istruzioni. Incredibile, fa parlare, cantare il tuo Commodore 64 poi programmarlo a fin che conosca la tua voce e ti risponda.

MODIFICA MPS 802 NEW GRAPHIC PLUS
L. 35.000
Eccezionale rende 100% compatibile la tua MPS 802 con tutti i programmi di grafica come (KOLA, PRINT SHOP GEOS ecc...) semplicissima da montare con chiavi istruzione in italiano!!!

EPROM NEW GRAPHIC MPS 801
L. 25.000
Si sostituisce al generatore di caratteri della stampante MPS-801 (per migliorare la leggibilità della scrittura car discendenti).

DISPONIBILI TUTTI I PEZZI DI RICAMBIO COMMODORE 64 Sconti particolari per rivenditori e quantitativi TELEFONATE! Per ulteriori informazioni richiedete i cataloghi per il vostro computer specificando il settore, inviando L. 1.000 in francobolli. Ricorda che alla **NEVEL** trovi anche tutto per COMMODORE AMIGA 64-128 MSX, SINCLAIR ZX & QL, ATARI ST e PC compatibili

VIDEODIGITAL 64
Nuovo digitalizzatore in cartuccia, digitalizza le tue più belle immagini con l'aiuto di una telecamera o videoregistratore semplicissimo da usaare con manuale in italiano.
Inoltre è possibile modificare le immagini con il KOALA ecc. **L. 70.000**

CARTRIDGE 80 COL. — **L. 39.000**
Permette di visualizzare le 80 colonne sul 64.

STARDOS NEW! Eccezionale novità un velocizzatore che supera persino la velocità dello speed-dos attiva i tasti funzione ecc. In una sola Eprom kit da inserirsi nel c64 con manuale in ital. Non necessita di elaborazioni al drive ne del cavo parallelo.
L. 39.000

Finalmente ! ecco
l'assistenza che cercavate

1 RAPIDA non più mesi ma solo pochi giorni !
2 SICURA perchè fatte da chi di computers se ne intende
2 AFFIDABILE lunga esperienza maturata nel settore

ASSISTENZA TECNICA a
COMPUTERS e PERIFERICHE COMMODORE

NOVITÀ L. 79.000

THE CHARTRIDGE È ...
... la rivoluzionaria cartuccia che vi permetterà di fare tutto ciò che è impossibile con le altre cartucce!
... dotata di un potente monitor interno per ogni vostra necessità
... compatibile al 100% con i nuovi drive Commodore e con lo Speed Dos!
... facilissima da usare e non richiede software!
... velocissima nei trasferimenti, fino a dieci volte più veloce di tutte le altre cartridges!
... dotata di un fast loader per disco, carica, salve e formatta in modo turbo!
Ma **THE CARTRIDGE** è molto, molto di più, da oggi, ad un prezzo imbattibile, la cartuccia imbattibile'
THE CARTRIDGE È ... la più potente cartuccia in commercio!!!

personalizzati compaiano nella cella in alto a sinistra dello schermo grafico, non è possibile sfruttare il comando CHAR, ma bisogna trasferire i dati relativi al carattere dalla mappa alla cella interessata.

Ad esempio, avendo ridefinito il carattere "C", ci si trova nella necessità di stamparlo nello schermo grafico.

Il carattere "C" corrisponde a D/CODE 03, e supponendo che l'inizio della mappa sia a 6000 (caso del C/16, vedi C.C.C. N.33), l'inizio delle descrizioni di "C" sarà posizionato a partire da 6000+(3*8)-1.

Per calcolare la fine delle variabilii e del programma basic, è necessario digitare in modo diretto...

Print peek(45)+peek(46)*256
...(per fine programma Basic) e...
Print peek(47)+peek(48)*256
...per la fine delle variabili.

E' possibile cancellare le lettere sbagliate con il tasto Inst/del.

Il programma, poichè agisce in Basic e per giunta su singoli pixel, risulta piuttosto lento, soprattutto quando i caratteri per riga, o per colonna, sono in numero esiguo; per evitare dubbi sulla reale attività del computer, è stato inserito il comando Sound (riga 330); nel caso del C/16 è opportuno cambiare il calcolo C=C*1.05 in C=C*1.01, perchè il range della frequenza è diverso da quello usato nel C/128.

Micro Text-editor

Il programma, che non ha nessuna pretesa di essere un word processor professionale, risulta piuttosto versatile: impartito il Run, digitate le frasi che desiderate vengano riprodotte sulla stampante tenendo presente che la stampa avviene superando il 74mo carattere. Se, poi, volete centrare automaticamente la riga di testo, premete Home prima del tasto Return.

Per scrivere una data in fondo alla riga, scrivetela come di consueto, poi premete ESC seguito da Return.

E' possibile cancellare le ultime 75 lettere digitate.

In alto a sinistra nel video è costantemente presente il numero di caratteri della stringa, non ancora "scaricata" su stampante.

Se arrivate a 75 caratteri con una parola a metà, non disperate, perchè il programma è predisposto per non stampare le parole che non verrebbero ospitate intere alla fine del rigo stesso.

```
100 REM ************************************
110 REM *         MINI TEXT EDITOR         *
120 REM *     PER C/16, PLUS/4, C/128      *
130 REM ************************************
140 X1$-CHR$(19)+CHR$(17):X3$-CHR$(166)+CHR$(157)
150 PRINT"1- C/16, PLUS/4":PRINT"2- C/128"
160 GET A$:IFA$-""THEN160
170 UU-3072:IFA$-"2"THENUU=1024
180 SCNCLR:OPEN4,4,7:PRINT#4,CHR$(15):PRINTCHR$(14)CHR$(8);X1$;
190 C-LEN(B$):C$-MID$(STR$(C),2,2):POKEUU,ASC(MID$(C$,1,1))
200 POKEUU+1,ASC(MID$(C$,2,1))
210 GETKEYA$
220 REM CURSORE: SU, GIU', DESTRA, SINISTRA, INSERT, CLR/HOME
230 IFA$-"⊐"ORA$-"▣"ORA$-"▮"ORA$-"▌"ORA$-"▐"ORA$-"⊐"THEN190
240 IFA$-CHR$(34)THENPRINTCHR$(34)CHR$(20);
250 IFA$-CHR$(19)THENCOLOR4,6,5:GOTO400
260 IFA$-CHR$(27)THENCOLOR4,9,4:GOTO370
270 IFA$-CHR$(20)THEN310
280 IFA$-CHR$(13)THENPRINTA$:GOSUB320:GOTO190
290 B$-B$+A$:IFLEN(B$)-75THENGOSUB330:PRINTA$;:ELSEPRINTA$+X3$;
300 GOTO190
310 IFLEN(B$)<1THEN190:ELSEB$-LEFT$(B$,LEN(B$)-1):PRINTA$+X3$;:GOTO190
320 PRINT#4,B$:B$-"":RETURN
330 FORI-75TO1STEP-1
340 IFMID$(B$,I,1)-CHR$(32)THENGOSUB430:RETURN
350 NEXTI:RETURN
360 PRINT#4,B$:B$-C$:RETURN
370 IFLEN(B$)>79THEN190
380 C$-LEFT$(B$,LEN(B$)):B$-"":FORI-1TO80-LEN(C$):B$-B$+" ":NEXT
390 B$-B$+C$:GOTO190
400 IFLEN(B$)>95THEN190
410 C$-LEFT$(B$,LEN(B$)):B$-"":FORI-1TO(80-LEN(C$))/2:B$-B$+" ":NEXT
420 B$-B$+C$:GOTO190
430 C$-RIGHT$(B$,LEN(B$)-I):B$-LEFT$(B$,I):GOSUB360:RETURN
```

COME STUDIARE GIOCANDO CON IL COMPUTER

Un gioco di simulazione per il C/64 che richiama alcune leggi della fisica

di **Maurizio Dell'Abate**

"Un corpo, lanciato in direzione non verticale assume (nel vuoto) una traiettoria parabolica".

E' questa una legge fondamentale della balistica, che studia il moto dei priettili.

Le caratteristiche della traiettoria variano in funzione di due fattori: l'angolo di inclinazione, con cui il corpo viene lanciato, e la velocità iniziale. Con queste semplici informazioni è possibile calcolare, tra le altre cose, a quale distanza dal punto di lancio il proiettile toccherà terra.

Il gioco

Scopo del gioco è colpire il bersaglio-carattere, che si trova sulla 25-ma riga del video, con un proiettile, che viene sempre lanciato con inclinazione di 45 gradi, ma da una posizione sempre più elevata.

La velocità iniziale (in km/h) viene digitata dal giocatore prima di ogni lancio. E' intuitivo che un corpo lan-

SCHEDA TECNICA

Gioco didattico di simulazione.

Hardware richiesto: C/64; non adattabile ad altri computer Commodore

Consigliato ai principianti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

ciato da un punto alto tocca terra in posizione più avanzata di uno lanciato, invece, da un punto basso (dotato di medesima velocità iniziale).

Quando un bersaglio viene colpito, il programma ne pone uno nuovo, ma in posizione differente. Terminati i tentativi a disposizione, viene reso noto il numero di successi, oltre al commento personale del computer.

Il proiettile è costituito da uno sprite formato da soli quattro dot: il bersaglio, invece, da un carattere semigrafico. E' quindi molto semplice rilevare la collisione tra le due entità.

Compilando il programma si ottiene una maggiore velocità di elaborazione, peraltro accettabile anche nella versione Basic.

Calcolo della traiettoria

Per chi voglia capire come viene calcolata la traiettoria del proiettile, forniamo qui di seguito le necessarie indicazioni (lasciate ogni speranza, voi che entrate).

Il vettore che simboleggia la velocità iniziale risulta sempre inclinato (per semplicità di programmazione) a 45 gradi; può essere idealmente scomposto in due vettori perpendicolari tra loro, giacenti sugli assi cartesiani.

Se il vettore originario ha modulo X, ciascuno dei due vettori ottenuti avrà modulo X/sqr(2), in quanto lato di un quadrato di diagonale X.

Durante lo spostamento, la proiezione del corpo sull'asse X determina l'avanzamento verso destra con moto uniforme, con velocità corrispondente al vettore ottenuto.

Se, invece, osserviamo la proiezione sull'asse verticale, notiamo che questa sale con velocità iniziale corrispondente al vettore ottenuto, ma decelera uniformemente, per effetto della gravità, fino a fermarsi nel punto più alto; immediatamente, sempre per effetto della gravità, inizia la discesa in moto uniformemente accelerato, fino a raggiungere l'asse orizzontale con la stessa velocità iniziale.

Abbiamo visto quindi come la traiettoria parabolica possa essere scomposta in due spostamenti ortogonali: in questo modo il programma pubblicato calcola, istante per istante, le coordinate dello sprite-proiettile.

```
100 REM GIOCO BALISTICO
110 REM BY M. DELL'ABATE 1988
120 :
130 G=9.8: FZ=.08
140 :
150 REM G  = ACCELERAZIONE DI GRAVITA'
160 REM FZ = FINEZZA DEL MOVIMENTO
170 :
180 :
190 POKE832,224:POKE835,224:POKE838,224:REM DEFINISCE LA FORMA DELLO SPRITE
200 POKE2040,13:REM I DATI PER LO SPRITE 0 SI TROVANO DA 832 (13*64) IN POI
210 V=53248:REM INDIRIZZO DI PARTENZA PER LA SCHEDA VIDEO
220 POKEV+21,1:REM ATTIVA LO SPRITE
230 POKEV+32,6:POKEV+33,0:REM SCHERMO NERO, BORDO BLU
240 FORAL=0TO100STEP5:REM CALCOLA LO SPOSTAMENTO IN ALTO DEL PUNTO DI PARTENZA
250 POKEV,24:POKEV+1,247-AL
260 IFCTTHEN290
270 PRINTCHR$(147):POKE211,0:POKE214,23:SYS58640:REM CURSORE SULLA RIGA 23
280 PRINT:PRINTTAB(25*RND(1)+4);"◆";:REM SHIFT+Z
290 PRINTCHR$(19):INPUT"VELOCITA' INIZIALE KM/H";VI:VI=VI*5/18:REM KM/H IN M/S
300 IFVI<=0THEN290
310 PRINTCHR$(19):FORI=0TO39:PRINTCHR$(32);:NEXT:REM CANCELLA LE SCRITTE
320 :
330 REM CALCOLO DELLA TRAIETTORIA
340 :
350 VS=VI/SQR(2):T=-FZ
360 RG=-G/2:CC=PEEK(V+31)
370 T=T+FZ
380 X=VS*T+24
390 Y=RG*T↑2+X-24+AL
400 IFX>255ORY<-5THEN450
410 IFY>247THEN370
420 IFPEEK(V+31)THEN460
430 POKEV,X:POKEV+1,247-Y
440 GOTO370
450 CT=1:GOTO490
460 CT=0:FORI=0TO50:POKE54296,15
470 POKE54296,0:POKEV+33,IAND1:NEXT
480 BC=BC+1
490 NEXT
500 :
510 REM FINE PARTITA
520 :
530 POKEV+21,0:REM SPRITE OFF
540 PRINTCHR$(147):REM CANCELLA SCHERMO
550 PRINT"HAI COLPITO";BC;"BERSAGLI."
560 PRINT
570 IFBC>=16THENA$="OTTIMO!":GOTO620
580 IFBC>=10THENA$="BUONO!":GOTO620
590 IFBC>=5THENA$="MODESTO...":GOTO620
600 IFBC>=1THENA$="MALE!":GOTO620
610 A$="CONSIDERA L'EQUITAZIONE..."
620 PRINTA$:PRINT
630 PRINT"PREMI UN TASTO PER GIOCARE ANCORA...":POKE198,0:WAIT198,1:POKE198,0
640 RUN:END

READY.
```

AMIGA: IMPARIAMO LE LINGUE

Come usare i file batch di AmigaDOS e la RAM disk per realizzare i nostri vocabolari personalizzati

di **Roberto Ferro**

Nello studio di una lingua straniera un problema che si presenta sovente è la memorizzazione di tutti i nuovi vocaboli di cui veniamo a conoscenza.

Spesso, infatti, ci ritroviamo a consultare il vocabolario per ricercare parole che, magari, abbiamo già incontrato, ma di cui non ricordiamo il significato o l'esatto modo in cui vanno scritte.

Una pratica applicazione, che unisca l'utile al dilettevole, potrà essere svolta dal nostro fantastico Amiga; vediamo come.

Abbiamo già esaminato, sul N. 47 di C.C.C, le possibilità che il CLI di Amiga offre riguardo la creazione di file Batch, ed abbiamo potuto osservare come sia fattibile la creazione di brevi programmi scritti con i soli comandi del DOS che semplifichino enormemente il lavoro quando si operi con i dischi.

Ciò su cui invece non ci eravamo soffermati, è la possibilità di sfruttare i file di comando per scopi che esulino quasi totalmente dalle operazioni di copiatura, cancellazione e trasferimento dei file del disco; ci riferiamo, appunto, alla gestione rapida e dinamica di un vocabolario.

Ci soffermeremo ora a considerare alcuni aspetti non esaminati nel precedente articolo sui file batch, approfondendo le possibilità che RAM disk e File Batch offrono.

Premettiamo, prima di proseguire, che il file ed i comandi di cui parleremo si riferiscono alla versione 1.2 del Workbench, per cui non ci sono problemi per i possessori di Amiga 500 e 2000, mentre per l'Amiga 1000 è necessario utilizzare la versione 1.2 sia del Kickstart che del Workbench, onde evitare inconvenienti di vario genere.

Ricordiamo che per l'uso del programma è sufficiente un solo disk drive.

Digitiamo i file

Come è possibile notare, i file necessari alla gestione del vocabolario sono due: Starter e CLI-Voc. Ciò è

dovuto al fatto che si lavora totalmente in RAM disk ed è necessario che questa sia "libera" per contenere il file stesso ed i comandi di cui ha bisogno.

Il primo command file (Starter Cli-Voc) predispone Amiga ad accogliere CLI-Voc nella RAM-disk.

Supponendo di voler salvare il primo file, quello più corto, sul disco MieiProgrammi con il nome Starter, non dovremo far altro che:

- Entrare in CLI
- Impartire ED MieiProgrammi: Starter
- Digitare il file
- Uscire dall'editor e salvare il programma con ESC e X

La stessa procedura va eseguita per salvare il batch del vocabolario vero e proprio tenendo presente che deve essere memorizzato sullo stesso disco su cui c'è lo starter ed essere obbligatoriamente chiamato CLI-Voc per questioni di "richiamo".

Il motivo che il file principale si debba chiamare per forza con questo nome, infatti, si spiega perchè esso si auto-lancia ed il nome usato per la procedura descritta è proprio "CLI-Voc".

Terminata la fase di digitazione e salvataggio potremo utilizzare il vocabolario impartendo...

EXECUTE MieiProgrammi:Starter

...che provvederà, dopo aver sistemato a dovere il RAM Disk, a lanciare il file del vocabolario vero e proprio.

Vediamo ora le modalità operative.

CLI-Voc

Dopo aver dato il via allo starter verrà richiesto di effettuare qualche scambio di disco per portare i comandi in RAM; in seguito apparirà il messaggio...

Inserisci il disco - Attendo 15 secondi...

...che ordinerà di inserire il disco su cui abbiamo salvato i file e su cui CLI-Voc catalogherà i nuovi vocabolari.

Terminata anche questa fase possiamo finalmente utilizzare il nostro file. Il prompt a cui dovremo rispondere con dei comandi sarà il seguente...

Lingua,Comando,(Vocabolo):

Abbiamo a disposizione quattro diversi comandi:

(L)ista delle parole
(C)erca una parola
(I)nserisci nuovi vocaboli
(L)ista (V)ocaboli (D)isco.

In breve: L, C, I, LVD.

CLI-Voc conserva in ordine alfabetico le parole immesse in un file sul RAM Disk che chiamerà con il nome della lingua di cui stiamo catalogando i vocaboli.

Alla prima comparsa del prompt dovremo indicare la lingua di cui vogliamo tenere un archivio di parole e, di seguito (separandolo con uno spazio) un comando fra quelli visti sopra.

Ad esempio...
Lingua,Comando,(Vocabolo): Inglese I

...seguito dal tasto return provocherà la creazione di un file di nome "Inglese" in cui immettere i vocaboli e darà una lista di quanto è presente nel file che, ovviamente, risulterà vuoto la prima volta che si utilizza il "programma".

E' importante sottolineare come la lingua DEVE essere indicata SOLO la prima volta che impartiamo un comando; in seguito sarà necessario solo il comando e niente altro.

I comandi

Abbiamo detto che i comandi sono quattro, ed in particolare:

- L - Serve ad ottenere una lista delle parole catalogate fino a quel momento;
- I - E' utile per inserire nuovi vocaboli. Verrà visualizzata una linea del tipo...

Inglese...............Italiano

...in testa alla quale dovremo inserire il vocabolo Inglese ed il corrispondente significato in italiano.

Le varie linee vanno separate con un Return; per concludere dovremo premere, dopo l'ultimo Return, CTRL e, contemporaneamente, lo slash presente accanto al tasto di backspace.

- C - Grazie a questo comando è possibile effettuare un'operazione di ricerca all'interno del file di vocaboli. Questo è l'unico comando che richiede un vocabolo (o parte di esso) digitato di seguito.

Se vogliamo, ad esempio, cercare la parola "casa" impartiremo...

c casa

...e, se presente, verrà visualizzata la linea contenente la parola "casa" con la rispettiva traduzione.

- LVD - Questo comando serve nel caso sia già presente su disco una lista di vocaboli; in caso affermativo ne verrà mostrata la lista.

E' importante sottolineare come la lista non venga caricata in memoria ed unita a quella che stiamo creando, ma semplicemente mostrata.

Per terminare le operazioni col vocabolario dovremo semplicemente impartire un Return a vuoto. In questo modo faremo sì che il file di vocaboli in memoria venga unito e riordinato alfabeticamente con uno eventualmente presente sul disco e poi registrato.

Se non esiste un file con il nome della lingua che abbiamo usato, ne verrà creato uno ex novo.

CLI-Voc non prevede un'opzione di correzione delle parole; se ciò dovesse rendersi necessario dovremo compiere "manualmente" l'operazione entrando nel file contenente le parole con...

Il RAM Disk di AmigaDOS

Lavorando con AmigaDOS, una delle cose senz'altro più comode ed utili è avere a disposizione il RAM Disk. Con questo è possibile compiere una varietà di operazioni che altrimenti risulterebbero lunghe e noiose, se non addirittura impossibili.

Ciò che rende così semplice e rapido il suo utilizzo è proprio la particolare struttura, che ora esaminiamo velocemente.

Innanzitutto bisogna dire che il RAM Disk è una porzione di memoria a grandezza variabile che Amiga gestisce come se fosse un'unità a dischi. Il sistema vede questo disco come uno dei device accessibii da DOS e possiamo renderci conto di ciò semplicemente digitando...

ASSIGN

...e vedendo scorrere l'elenco completo dei device di AmigaDOS compreso "RAM:". Abbiamo detto che il device RAM si comporta esattamente cose se fosse un disco, sempre inserito, di nome "RAM:"; inoltre la sua capacità è sempre relativa a quanto vi è contenuto: se abbiamo un file di 200 K sul RAM Disk, quest'ultimo occuperà esattamente 200 K nè uno di più nè uno di meno.

A conferma di ciò è sufficiente digitare, da CLI, il comando...

INFO

...per essere edotti su quanto accade nei drive collegati all'Amiga, compreso il RAM Disk.

Ci si potrà anche accorgere del fatto che il disco della RAM è sempre pieno al cento per cento. Questo modo di operare è estremamente efficiente perchè non provoca spreco di memoria. Su sistemi operativi quali l'MS-DOS, ad esempio, è invece necessario dichiarare che si vuole utilizzare una porzione di memoria RAM da gestire come un disco; in seguito è anche necessario specificare quanto si vuole sia grande questa zona; nonostante, poi, nel RAM Disk non ci sia un bel niente, verranno sottratti, dalla memoria centrale, i Kilobyte indicati.

Tutto ciò non accade su Amiga che non ha bisogno di alcuna dichiarazione per accedere al disco virtuale.

Un'altra caratteristica molto importante è l'alta velocità di trasferimento dati. Quando infatti operiamo con file o comandi presenti sul RAM Disk ci accorgiamo piacevolmente di come le nostre operazioni vengano portate a termine con estrema rapidità.

Il command file CLI-Voc non è che un altro esempio di come possa tornare utile tale velocità.

Il RAM Disk è disponibile anche per le operazioni da compiere tramite WorkBench. Per farne apparire l'icona si dovrà impartire...

DIR Ram:

I possessori di Amiga 500 e 2000 potranno verificare come nella Startup-Sequence dei loro Workbench questo comando sia già incluso provocando la comparsa del disco RAM: sopra a quello del WorkBench stesso, subito dopo l'inizializzazione.

E' ancora molto importante sottolineare che la presenza di un disco virtuale può tornare utile in molti casi come ad esempio quando vogliamo sperimentare l'effetto di un certa procedura prima di applicarla sul disco reale.

Infine rimane da dire che tutto il contenuto del RAM Disk viene perso allo spegnimento o alla reinizializzazione della macchina.

ED MieiProgrammi:Inglese

...e dopo aver corretto la parola sbagliata uscire con ESC e X.

Struttura di CLI-Voc

Abbiamo già accennato come una delle caratteristiche salienti di CLI-Voc sia quella di essere gestito completamente in RAM Disk. E' facile capire che un simile modo di operare comporta enormi vantaggi in velocità e precisione nel lavorare. Inoltre, evitando scambi di dati con il disco magnetico, otteniamo una minore usura del disco stesso (ed una maggiore affidabilità...).

Procediamo, però, con ordine.

Esaminando il file Starter vediamo come questo provveda a liberare il RAM disk e ad immettervi CLI-Voc ed i comandi di cui ha bisogno, dopodichè si occupa di dare il via allo stesso CLI-Voc.

In questo file possiamo notare l'uso del carattere punto e virgola (;) che ha la stessa funzione svolta dall'istruzione REM del BASIC. Per il resto non c'è nulla di particolare tranne i simboli usati nel comando di DELETE.

Essi rappresentano un pattern matching: particolari simboli usati per dire al sistema che nell'eseguire un comando deve considerare tutti i nomi con determinate caratteristiche. Nel nostro caso il pattern ha la stessa funzione che sui piccoli Commodore come il C/64 e su altri sistemi ha l'asterisco (*): serve a prendere in considerazione qualunque nome di file presente nel disco da noi indicato. E' facile quindi capire che con il comando DELETE, riferito al pattern matching, provvediamo a liberare il RAM Disk di tutti i file, per permettere al nostro vocabolario di trovare lo spazio sufficiente.

E' importante sottolineare come i vari comandi del CLI che si riferiscono al dispositivo RAM: abbiano influenza solo sul RAM Disk e non su

tutta la RAM di Amiga, pertanto è necessario, a meno che non possediate un'espansione, che nessun altro programma giri contemporaneamente (multitasking) al vocabolario, onde evitare spiacevoli messaggi di GURU MEDITATION od altro dovuti all'esaurimento della memoria.

Il programma principale

Esaminando CLI-Voc possiamo subito notare la presenza di .Key all'inizio.

Abbiamo già esaminato sul N. 47 che questo comando serve a dichiarare "variabili" il cui valore viene immesso al momento del lancio del file stesso.

Tali variabili possono essere utilizzate all'interno del file semplicemente racchiudendole tra i segni "maggiore" e "minore".

Scorrendo attentamente il file ci accorgiamo anche dell'uso dell'asterisco (*) come se fosse un file. In effetti con l'asterisco indichiamo una periferica, e più precisamente il video. Esso viene trattato come se fosse un file ed infatti il comando JOIN (inserito verso l'inizio del programma) serve ad unire il contenuto del video con quello del file "lingua" mettendo il tutto in "LinguaRam", e ciò è proprio quello che avviene quando usiamo il comando "I" di CLI-Voc.

In parole povere l'asterisco rappresenta il video così come DF0: rappresenta il drive interno e PRT: la stampante.

Un altro paio di comandi molto interessanti che troviamo sono SEARCH e SORT. Su questi due comandi si basa l'intero funzionamento del vocabolario.

Con il primo, nel nostro caso, compiamo la ricerca di una parola data all'interno di un file, mentre con SORT riordiniamo alfabeticamente (secondo il codice ASCII) un file di testo e poniamo la lista ordinata in un secondo file.

In CLI-Voc sono stati anche ampiamente sfruttate le strutture IF/ELSE/ENDIF e i salti ed etichette come SKIP e LAB. Possiamo anche vedere che in un paio di casi le condizioni sono state nidificate a testimonianza delle ampie possibilità di programmazione che il CLI di Amiga offre (sempre considerando che stiamo trattando con comandi DOS e quindi non specificatamente pensati per la programmazione).

Infine non rimane che considerare l'ultimo comando di CLI-Voc, cioè EXECUTE.

Con questo comando provochiamo l'autostart del batch, e fin qui niente di particolare; ciò su cui invece porremo la nostra attenzione è il punto interrogativo(?).

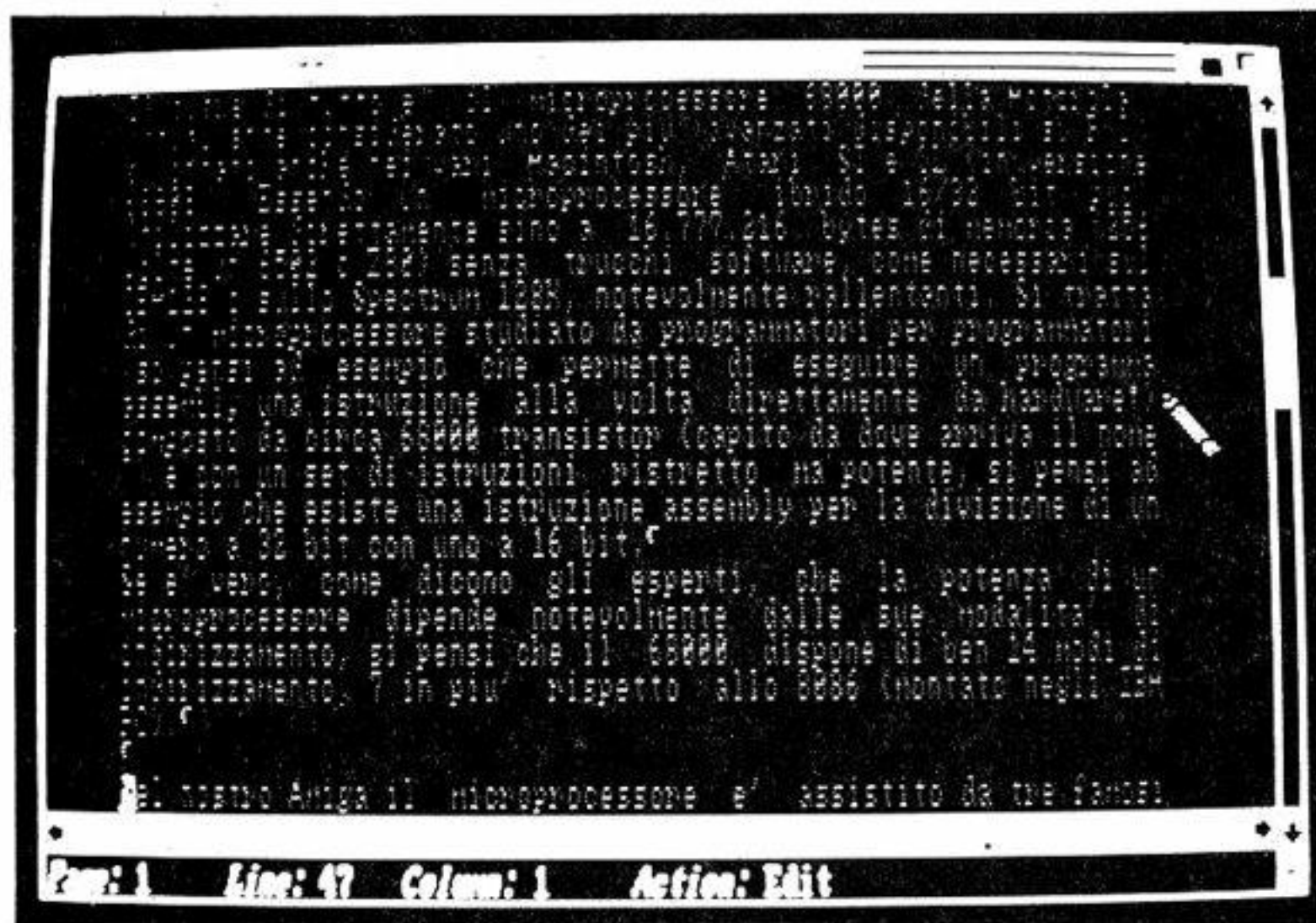
Bisogna sapere che quando da CLI impartiamo un comando seguito solo da un punto interrogativo provochiamo la comparsa della lista e del formato dei parametri richiesti da quel comando (Argument Template) in attesa che vengano digitati. Un discorso analogo vale per i file di comando: nel caso questo preveda parametri, sarà sufficiente far seguire, al nome del file, un punto interrogativo affinchè vengano mostrati i parametri richiesti.

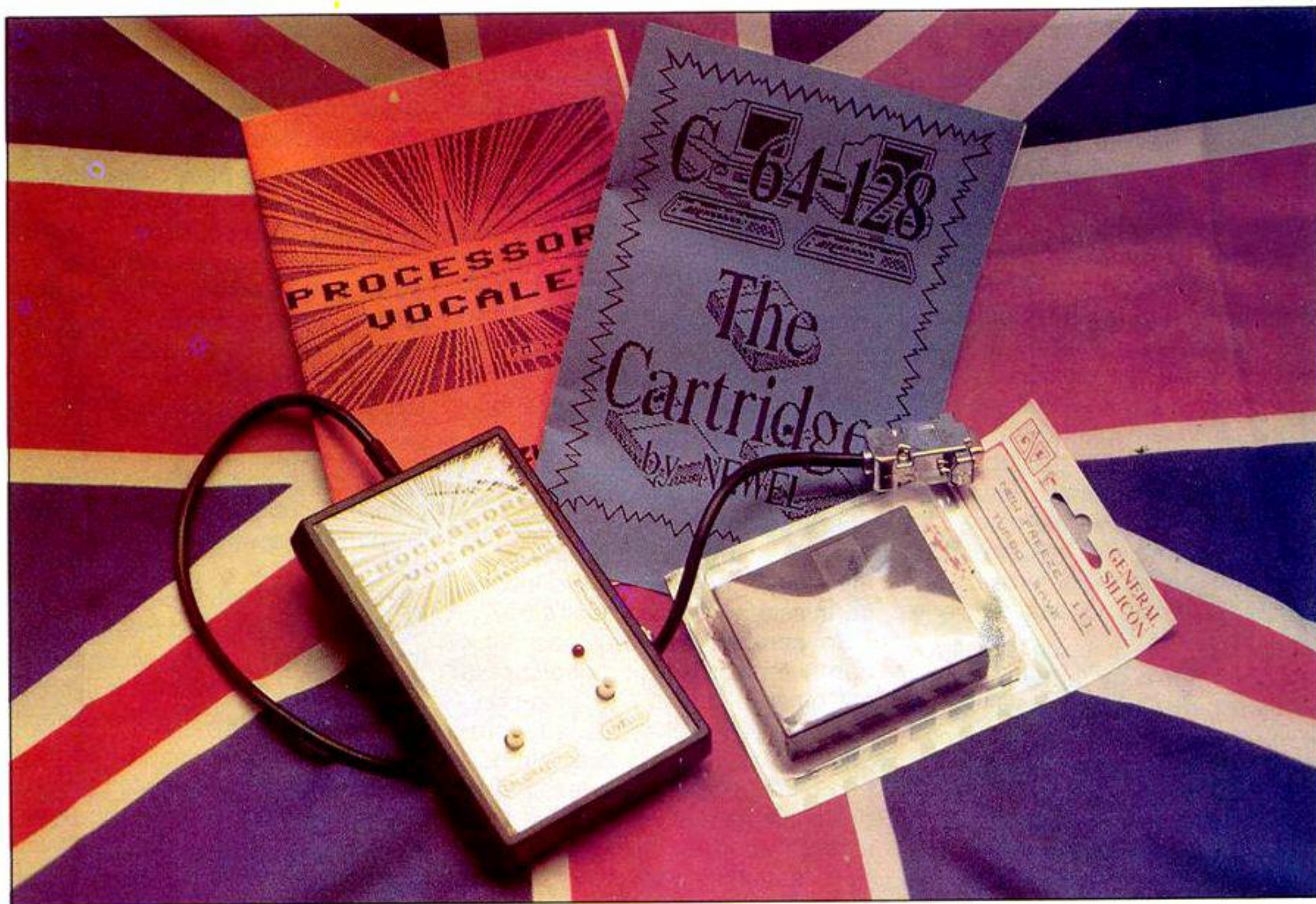
Questa caratteristica è stata sfruttata in CLI-Voc per provocare la comparsa del prompt che indica i parametri da immettere.

Se, infatti, impartiamo...

EXECUTE ram:CLI-Voc ?

...avremo la comparsa dei parametri indicati nel file, di seguito a .Key. E' stato fatto uso del punto interrogativo sin dallo Starter che lancia CLI-Voc con un comando come quello appena visto. In questo modo si spiega anche perchè è necessario che il nome della lingua che trattiamo sia inserito solo la prima volta in cui impartiamo un comando (ma tutte le volte che lanciamo da capo il vocabolario): l'EXECUTE finale del vocabolario provvede già ad immettere la lingua e quindi soddisfa già il primo parametro di .Key.

Peccato che il sistema non "sente" se un parametro è già stato immesso e mostra sempre l'intera lista dei parametri presenti. Precisiamo che il parametro vocabolo è posto tra parentesi solo per indicare che non sempre è richiesto e non perchè questa sintassi abbia particolare significato agli occhi di Amiga.

```
; ************************************
; *          Starter Cli-Voc          *
; *              4/1/88               *
; *         di Roberto Ferro          *
; ************************************
;
;
ASSIGN c: sys:c                   ; Assegna al device C:
PATH RESET                        ; la directory c del WB
DELETE ram:#? ALL QUIET           ; Cancella tutto il
MAKEDIR ram:c                     ; RAM Disk e vi
COPY c: TO ram:c ALL QUIET        ; porta i comandi
ASSIGN c: ram:c
CD ram:
;  Trasferisce "CLI-Voc" in RAM disk
ECHO "Inserisci il disco - Attendo 15 secondi..."
WAIT 15
COPY Df0:CLI-Voc TO ram:
;  Lancia il file "CLI-Voc"
ECHO "Indicare la lingua solo al primo comando."
EXECUTE CLI-Voc ?
```

```
; ************************************
; *          CLI-Voc V1.3            *
; *             4/1/88               *
; *       di  Roberto  Ferro         *
; ************************************
;
; Immissione parametri
.key Lingua,Comando,(Vocabolo)
IF NOT EXISTS ram:<lingua>
  ECHO "Creo '<lingua>', premi CTRL + \."
  TYPE > ram:<lingua> *
ENDIF
; Esegue il comando ricevuto
IF "<comando>" EQ "i"
  ECHO "Separa i vocaboli con il RETURN"
  ECHO "Premi CTRL + \ per finire"
  ECHO "<lingua>:............Italiano"
  JOIN * ram:<lingua> AS ram:LRAM
  SORT ram:LRAM TO ram:<lingua>
  SKIP Fine
ENDIF
IF "<comando>" EQ "c"
  SEARCH FROM ram:<lingua> "<(vocabolo)>"
  ECHO "."
  SKIP Fine
ENDIF
IF "<comando>" EQ "l"
  ECHO " "
  ECHO " - Vocaboli di <lingua>"
  ECHO "."
  TYPE ram:<lingua>
  ECHO " "
  SKIP Fine
ENDIF
IF "<comando>" EQ "lvd"
  ECHO "Searching..."
  IF EXISTS Df0:<lingua>
   ECHO "Loading..."
   TYPE Df0:<lingua>
  ELSE
   ECHO "File <lingua> non presente"
  ENDIF
  SKIP Fine
ENDIF
IF "<comando>" EQ ""
  CD Df0:
  IF NOT EXISTS <lingua>
   COPY ram:<lingua> TO <lingua>
  ELSE
   JOIN ram:<lingua> <lingua> AS ram:LRAM
   SORT ram:LRAM TO <lingua>
   DELETE ram:LRAM
  ENDIF
  DELETE ram:<lingua>
  QUIT
ENDIF
; Riprende l'auto-esecuzione
; immettendo già la lingua
LAB Fine
EXECUTE ram:CLI-Voc <lingua> ?
```

E' PRIMAVERA, SPUNTANO LE CARTUCCE...

Una fresca ventata di novità nel campo, quanto mai vasto, degli accessori per il Commodore 64

di **Michele Maggi**

Per ciò che riguarda il fiorente mercato delle cartucce per C/64 (soprattutto sprotettrici), si affacciano di continuo nuovi prodotti, sempre più perfezionati e pieni di interessanti opzioni.

I nomi delle cartucce sono molto accattivanti, quasi fossero una promessa di veri e propri miracoli di sprotezione e "hacking".

Dalle prime cartucce, che trasferivano dalla memoria, su disco, più file con procedimenti a volte lunghi e dall'esito spesso incerto, siamo arrivati ad uno standard qualitativo che permette di trasferire ad una velocità notevolmente più elevata e con una probabilità di successo molto più alta che in passato.

E' questo il caso di "The Cartridge", un prodotto commercializzato dalla ditta Newel S.r.l. di Milano, che si presenta più come un laboratorio che come una cartuccia nell'accezione tradizionale del termine.

The Cartridge

Il termine "laboratorio" è stato scelto proprio in funzione di tutte le opzioni che fanno di questa cartuccia una vera e propria stazione di lavoro per hacker.

Dato per scontato che il lavoro principale (sprotezione e trasferimento di programmi) viene effettuato egregiamente, le altre caratteristiche più interessanti sono senz'altro quelle relative alla possibilità di visualizzazione, e modifica, sia di sprite che di videate in alta risoluzione, con facoltà, da parte dell'utente, di salvare separatamente dal programma ciò che più interessa.

Le potenzialità di "The Cartridge", naturalmente, sono numerose, e tra queste accenniamo alla funzione "sprite killer" che inibisce le collisioni sprite-sprite o sprite-fondo in modo da sopravvivere nei giochi in cui la "morte" è determinata dalla collisione con sprite od oggetti. Incluso nella cartuccia è presente anche un potente Fastload che permette di compiere operazioni di caricamento o salvataggio a velocità di molto superiore a quella normale.

Per usi particolari sono anche disponibili vari comandi Basic che svolgono diverse funzioni di solito non possibili con il Basic standard del C/64: Old, Merge, Bsave, Bload, Append, Auto e così via.

Infine è possibile agire direttamente sulla memoria grazie ad un Monitor Assembler che è dotato di tutti i comandi necessari per la programmazione in Linguaggio Macchina.

Frizzy MK3

Accanto a "The Cartridge", vengono commercializzati anche altri modelli di cartucce che svolgono la loro "missione" senza però offrire la tipologia di utility riscontrata in The Cartridge.

Una di queste cartucce è la "Frizzy MK3", evoluzione della serie di Freeze Frame. Come per The Cartridge, anche Frizzy MK3 non richiede software, ma è pronta non appena inserita nel computer.

Chi si trovasse nella necessità di acquistare una cartuccia, rimarrebbe disorientato di fronte alla marea di prodotti, in apparenza analoghi, che invadono il mercato; ciò si traduce molto spesso nella rinuncia all'acquisto o, peggio, nell'acquisto sbagliato.

La domanda più freqente, "perchè esistono tanti tipi di cartucce?" trova la sua naturale risposta nell'elevato e diversificato numero di protezioni che oggi vengono applicate ai giochi in commercio.

Ciò premesso, è evidente come possa nascere l'esigenza di possedere differenti cartucce, in modo da contare su tecniche di sprotezione differenti: "chissà se dove si inchioda questa non riesca quell'altra?".

Un parallelo con qualsiasi altro hobby è presto fatto: molto spesso chi gioca a tennis o a ping-pong possiede più di una racchetta, proprio per essere in grado di affrontare, nel migliore di modi, le varie sfide; analogamente chi coltiva l'hacking (= sportezione) come... hobby deve essere in possesso di tutti i "ferri del mestiere" che ritiene necessari.

Processore vocale

Chi invece ama il computer, ma per vari motivi (di ordine morale e non) non prova interesse per le sprotezioni, è molto spesso attratto da altri prodotti informatici, che possono in qualche modo impreziosire il software di cui si dispone.

Un classico esempio può essere fornito dal "Processore vocale" che, tramite il collegamento via microfono tra C/64 e mondo esterno, è in grado di sintetizzare, riprodurre e (addirittura) riconoscere la voce umana.

Le applicazioni di questo accessorio sono limitate solo dalla fantasia di chi lo usa, in quanto il campo di applicazione, come è facilmente intuibile, è assai vasto.

Allegato alla confezione è un manuale ben realizzato, di una quindicina di pagine, che spiega nei dettagli le operazioni da compiere.

Dopo la parte riguardante la taratura e le connessioni da effettuare per ottimizzare il livello di fedeltà di riproduzione, la prima cosa da provare è l'emozione di sentir riprodotta la propria voce dal computer.

Con l'opzione "Dimostrazione" è possibile infatti insegnare al computer una determinata parola, o breve frase, che poi sarà ripetuta con diverse velocità e perfino dotata, volendo, di un particolare effetto eco.

E' prevista anche la possibilità di creare un orologio ed una calcolatrice parlanti, tramite due utility presenti sul disco.

La parte più rivoluzionaria del pacchetto è quella riguardante il riconoscimento vocale, che avviene tramite un processo di "trainig" e la creazione di opportuni file-vocabolario su disco.

I risultati ottenuti durante la prova effettuata sono da considerare soddisfacenti, specialmente se si considera che il C/64 ha una quantità di memoria modesta, fatto che limita la fedeltà ottenibile durante l'operazione di campionamento.

Non ci si illuda, quindi, di poter "parlare" al computer e, magari, farsi comprendere! Il divertimento, tuttavia, è assicurato.

Particolari sezioni del manuale sono dedicate ai comandi Basic specifici, come Learn, Speak, Recogn, ed altri, che permettono una facile gestione dell'accessorio.

Particolare non trascurabile, alla confezione è allegato un microfono a morsa, da utilizzare anche per registrazioni musicali e sonore di vario genere.

Tutto il materiale necessario alle prove è stato gentilmente fornito dalla Newel, alla quale è possibile rivolgersi per entrare in possesso di maggiori informazioni:

NEWEL s.r.l.
Via Mac Mahon, 75 - Milano
Tel. 02/32.34.92 - 32.70.226

ATTENTI AI TASSI!

Gli argomenti presentati da alcune società finanziarie, per convincere i potenziali clienti a chiedere un finanziamento, nasconde, molto spesso, una specie di raggiro; vediamo quale

di **Antonio Pastorelli**

Non stiamo certo parlando di quei graziosi animaletti dei boschi, che, anzi, devono ben nascondersi dall'uomo per non diventare pennelli (come troppo spesso accade!).

I tassi di cui parliamo sono quelli finanziari.

In un paese industriale, quale è l'Italia, caratterizzato da intense attività economico-commerciali, la moneta assume un ruolo di primaria importanza, intorno alla quale si sviluppano altre attività e servizi (leasing, carte di credito, mutui, fondi d'investimento, eccetera).

Diventa sempre più frequente il ricorso a tali servizi in alternativa ai tradizionali mezzi di pagamento; pertanto bisogna saper valutare quanto effettivamente costa una determinata operazione; oppure stabilire se un certo servizio è prestato ad un costo equo, tenuto conto delle normali condizioni di mercato.

Il problema che ci porremo è proprio quello relativo ai mutui ed ai prestiti in generale.

Il programma pubblicato può essere utilizzato prima di acquistare un appartamento, contraendo un mutuo; oppure stabilire il reale costo di un'operazione finanziaria che ci vede comprare un'auto versando subito un anticipo e, la restante parte, in quote mensili.

Proprio in queste occasioni è bene

prestare attenzione ai "Tassi", dei quali parlavamo prima.

Un esempio pratico

Supponiamo di voler acquistare un'auto, che costa 18.000.000, e di pagare in contanti 8.000.000, mentre per i restanti 10 milioni contraiamo un prestito, al 10% annuo, con rimborso in 36 rate mensili (3 anni).

Quasi sicuramente il rivenditore farà un ragionamento del tipo:

• 10.000.000 + interessi al 10% per 3 anni (3.000.000) = 13.000.000

• Dividendo 13.000.000 (capitale + interessi) per il numero delle rate (36), otteniamo l'importo della rata mensile: 361.111.

IL LINGUAGGIO USATO

Il programma è stato scritto in Gw-Basic, con il supporto dell'emulatore Ms-Dos & Gw-Basic prodotto dalla Systems Editoriale, e pertanto gira correttamente solo su C-64 (o C/128 in modo 64), con Emulatore, oppure su computers MS-DOS compatibili.

Nonostante ciò, le modifiche da apportare per far girare il programma con altri linguaggi interpreti sono di una semplicità incredibile; suggeriamo ai lettori, quindi, la conversione del listato per adattarlo al proprio computer.

Si tenga presente, a tale scopo, che l'istruzione "Keyoff" serve solo per cancellare l'indicazione dei tasti funzione.

L'istruzione "Locate" consente di posizionare il cursore in una zona particolare del video.

Il comando "Beep" emette un breve impulso sonoro.

"Inkey$" costringe il computer ad attendere la pressione di un tasto.

"Cls", infine, cancella lo schermo.

Il ragionamento sembra non fare una piega, eppure c'è qualcosa che non va per il verso giusto.

Vi aiuteremo a scoprire che, con una rata mensile di 361.111, pagate, in realtà, un tasso di interesse pari al 19.465% vale a dire quasi il doppio di quello dichiarato!

Il motivo della differenza di tasso è duvuto al fatto che, calcolando gli interessi per tre anni su 10 milioni di prestito, si dovrebbe sottintendere che il pagamento dell'importo prestato venga effettuato, in un'unica soluzione, alla scadenza del prestito (nel nostro caso, dopo tre anni).

In realtà le cose, come è noto, vanno diversamente: i pagamenti vengono effettuati mese per mese, a partire da quello successivo alla stipulazione del contratto! Pertanto alla scadenza dei tre anni dovrebbe rimanere una sola rata da pagare e non l'intero importo.

In definitiva, nel caso di pagamenti periodici, bisognerebbe calcolare gli interessi sulle varie rate, applicati soltanto al periodo di tempo che intercorre dalla data di stipulazione del contratto fino al suo pagamento.

In altre parole, con la prima rata si dovrebbero pagare gli interessi relativi ad un solo mese, sulla seconda per due mesi e, solo sull'ultima, per tre anni.

Per risolvere il problema, quindi, è necessario seguire un altro ragionamento, pur se non così semplice come quello appena visto.

Si agisce in modo che la somma delle rate, scontate per il periodo di dilazione, fornisca esattamente la somma oggetto del prestito.

Per spiegare meglio il problema, ipotizziamo un caso estremamente semplice, in cui viene prestata la cifra di 10.000.000, rimborsabile in due rate annue (di 5.761.905 cadauna, cifra che verrà determinata in un modo particolare, come vedremo in seguito), al tasso del 10% annuo.

La situazione è la seguente:

Allo scadere della prima rata è necessario pagare l'interesse relativo ad un anno al 10%, somma che deve essere scorporata (scontata), per arrivare all'importo originario:

5.761.905 / 1.1 = 5.238.095

...cioè 10%

La seconda rata, invece, conterrà gli interessi relativi al periodo di due anni e pertanto deve essere scontata al 10% per tale periodo:

5.761.905 / 1.21 = 4.761.905

...cioè 10% annuo calcolato, però, con l'interesse composto.

La somma dei due importi è, appunto, 10.000.000.

Quest'ultimo esaminato, pertanto, dovrebbe essere il metodo "onesto" per restituire i 10 milioni al 10% effettivo annuo.

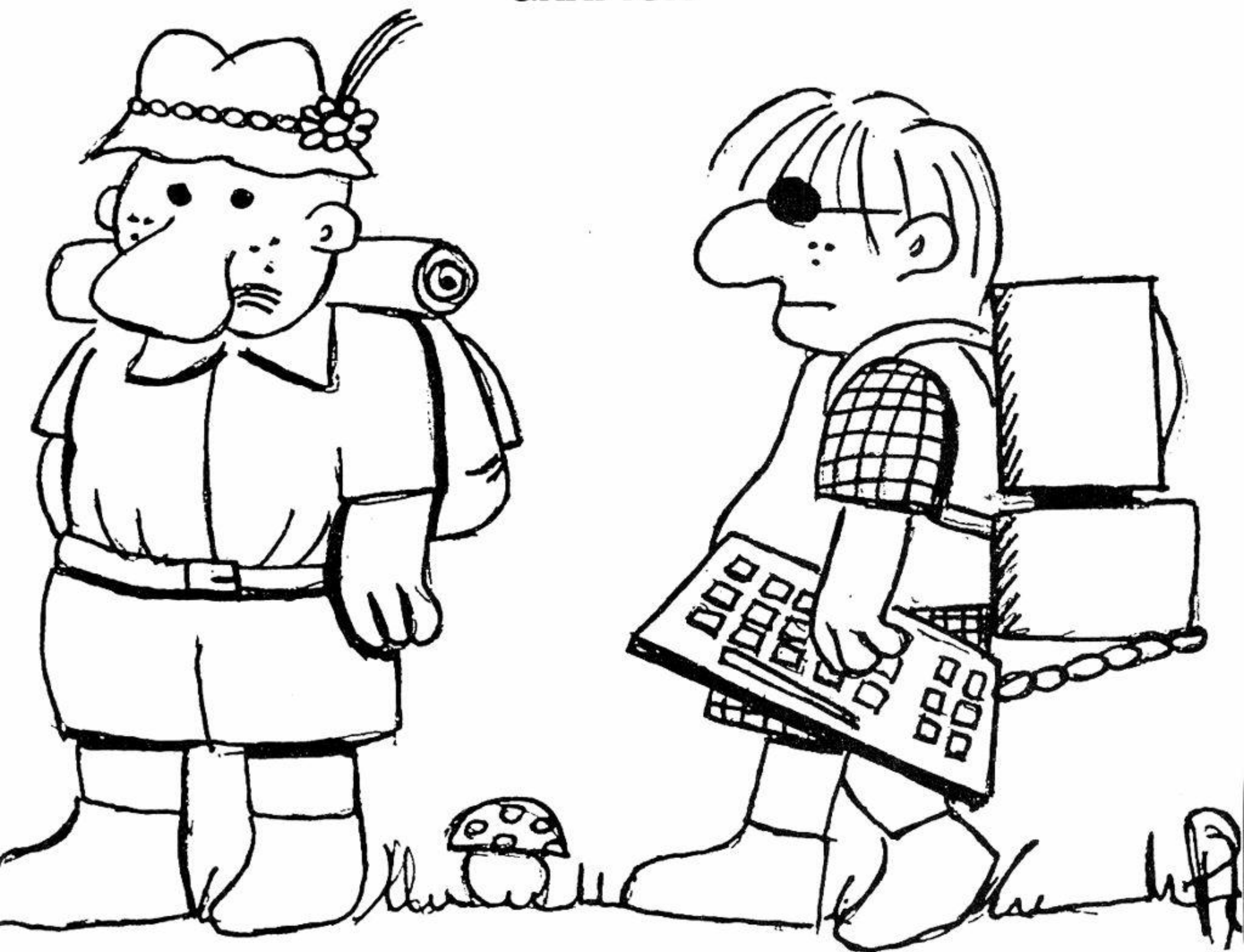
Il programma

Il programma "MUTUI", permetterà, con estrema semplicità, di calcolare correttamente l'importo della rata di un prestito, o di conoscere il "valore attuale" (cioè reale) di una rateazione.

La seconda opzione serve per conoscere l'importo che ha dato origine ad una serie di pagamenti, e si rivela particolarmente utile quando si voglia determinare l'importo da restituire in caso di estinzione anticipata di un prestito.

In tal caso basterà indicare al computer l'importo della rata, la periodicità dei pagamenti, il tasso, ed infine il numero di rate rimaste da pagare, ottenendo così l'importo da restituire per estinguere il contratto.

```
290 REM ************************
300 REM * AMMORTAMENTO  MUTUI *
320 REM *  PER: C-64+GWBASIC  *
340 REM ************************
360 REM *BY ANTONIO PASTORELLI*
370 REM ************************
380 :
390 KEYOFF:REM SU PC-IBM & COMPATIBILI DIGITARE "KEY OFF"
400 CLS:LOCATE 1,10:PRINT "AMMORTAMENTO  PRESTITI"
420 LOCATE 8,1:PRINT "1) CALCOLO RATA PER IL RIMBORSO DI UN"
440 LOCATE 9,4:PRINT "PRESTITO"
460 LOCATE 10,1:PRINT "2) CALCOLO VALORE ATTUALE DI UNA"
480 LOCATE 11,4:PRINT "RENDITA"
500 A$=INKEY$
520 IF A$="1" THEN 680
540 IF A$="2" THEN 2000
560 GO TO 500
580 :
600 REM ****************
620 REM * CALCOLO RATA *
640 REM ****************
660 :
680 CLS:LOCATE 1,6:BEEP
690 PRINT "CALCOLO RATA RIMBORSO PRESTITO":PRINT:PRINT
```

```
700 INPUT "IMPORTO PRESTITO";IM
720 INPUT "NUMERO RATE";NR:PRINT
740 PRINT "PERIODICITA': 1) MENSILE"
760 LOCATE 8,15:PRINT "2) BIMESTRALE":LOCATE 9,15:PRINT "3) QUADRIMESTRALE"
780 LOCATE 10,15:PRINT "4) SEMESTRALE":LOCATE 11,15:PRINT "5) ANNUALE":PRINT
800 A$=INKEY$
820 IF A$>="1" AND A$<"6" THEN 860
840 GOTO 800
845 :
846 REM *************************
847 REM *         CALCOLO       *
848 REM * PERIODIZZAZIONE ANNUA *
849 REM *************************
850 :
860 IF A$="1" THEN PE=12
880 IF A$="2" THEN PE=6
900 IF A$="3" THEN PE=3
920 IF A$="4" THEN PE=2
940 IF A$="5" THEN PE=1
941 :
942 REM *****************
943 REM * CALCOLO TASSO *
944 REM * PERIODALE     *
945 REM *****************
946 :
950 IF RT=1 THEN RT=0:RETURN
960 INPUT "TASSO ANNUO DI INTERESSE";TA
980 TP=1+TA/100:TP=TP↑(1/PE)
982 IF RT=1 THEN RT=0:RETURN
990 CO=(1-(1/TP)↑NR)/(TP-1)
992 IF RT=1 THEN RT=0:RETURN
1000 RA=IM/CO
1010 IF RA-INT(RA)>.5 THEN RA=INT(RA)+1:GO TO 1030
1020 RA=INT(RA)
1030 BEEP
1040 PRINT:PRINT "RATA: ";RA
1050 A$=INKEY$:IF A$="" THEN 1050
1060 RUN
1100 :
1110 REM *****************
1120 REM *  CALCOLO  DEL  *
1130 REM * VALORE ATTUALE *
1140 REM *****************
1150 :
2000 CLS:LOCATE1,2:BEEP
2010 PRINT "CALCOLO VALORE ATTUALE  DI UNA RENDITA":PRINT:PRINT
2020 INPUT "IMPORTO RATA";RA
2030 INPUT "NUMERO  RATE";NR
2035 PRINT
2040 RT=1:GOSUB 740:REM CALCOLO NUMERO RATE ANNUE
2045 INPUT "TASSO ANNUO";TA
2050 :
2060 RT=1:GOSUB 980:REM CALCOLO TASSO PERIODALE
2070 RT=1:GOSUB 990:REM CALCOLO COEFFICIENTE DI SCONTO
2080 IM=RA*CO
2090 IF IM-INT(IM)>.5 THEN IM=INT(IM)+1:GO TO 2110
2100 IM=INT(IM)
2110 PRINT:PRINT:BEEP
2120 PRINT "VALORE ATTUALE DELLE RATE:";IM
2130 GO TO 1050
```

UNA VECCHIA CONOSCENZA

Molti lettori hanno richiesto una routine di hard copy per la pagina grafica del C/128

La grafica, si sa, appassiona tutti gli utenti di computer, soprattutto quelli che posseggono il C/128 e che lo utilizzano, quasi sempre, in modo 40 colonne.

La routine che proponiamo è stata già pubblicata tempo fa su C.C.C. ("incorporata" in un programma grafico), ma la proponiamo nuovamente sotto una forma lievemente diversa, per la gioia dei neo-lettori.

E' numerata da 63000 in poi, in modo da agevolare un suo eventuale inserimento in programmi più ampi.

Naturalmente sarà necessario attivare la routine stessa (con un banale SYS 5392) tutte le volte che si desidera riversare su stampante (Mps-803 compatibile) una schermata grafica realizzata, ad esempio, con programmi in Basic 7.0, simili al demo riportato in queste stesse pagine.

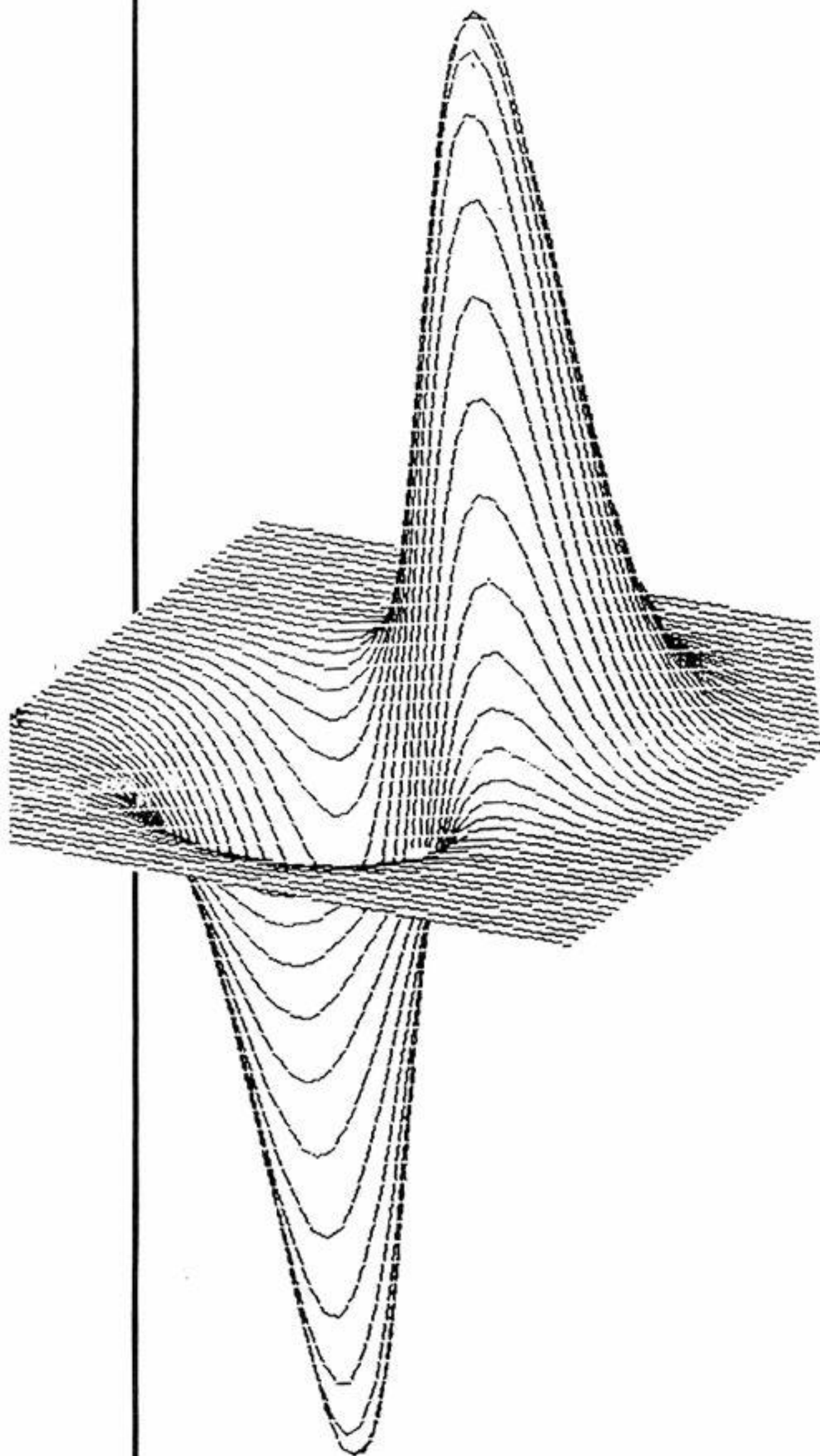
SCHEDA TECNICA

Software per applicazioni grafiche

Hardware richiesto: C/128 non adattabile ad altri computer Commodore

Richiede stampante MPS-803 o compatibili

Consigliato a tutti gli appassionati di grafica

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

```
100 REM DEMO HI-RES C/128
110 :
120 GRAPHIC 1,1
130 FORI=1TO 50
140 DRAW 1,0,0 TO 300,I*10
150 NEXT
160 GRAPHIC 0
170 PRINT"DIGITA: SYS 5392"
```

```
63000 REM SOFTWARE BY MARIO E STEFANO
63010 REM CIURLI DI ROMA
63020 :
63030 PRINTCHR$(147)
63040 PRINT"ROUTINE DI HARD COPY PER"
63050 PRINT"C/128 ALLOCATA DA"
63060 PRINT"5392 A 5740":PRINT
63070 PRINT"ATTIVABILE CON: SYS 5392"
63080 X$="ERRORE NEI DATA"
63090 FOR I=5392 TO 5740
63100 READ A: POKE I,A: X=X+A
63110 NEXT:IF X<>31578 THEN PRINT X$
63120 DATA 169,4,170,160,255,32,186,255
63130 DATA 169,0,32,189,255,32,192,255
63140 DATA 162,4,32,201,255,169,8,32
63150 DATA 210,255,169,13,32,210,255,169
63160 DATA 0,141,3,21,169,6,141,4
63170 DATA 21,169,0,141,0,21,141,1
63180 DATA 21,169,128,141,5,21,173,3
63190 DATA 21,141,2,21,173,0,21,141
63200 DATA 6,21,173,1,21,41,248,141
63210 DATA 7,21,169,0,141,8,21,141
63220 DATA 9,21,173,2,21,74,74,74
63230 DATA 170,240,23,169,0,168,72,24
63240 DATA 152,105,64,168,104,105,1,72
63250 DATA 202,208,244,140,9,21,104,141
63260 DATA 8,21,173,2,21,41,7,141
63270 DATA 10,21,173,1,21,41,7,141
63280 DATA 11,21,169,7,56,237,11,21
63290 DATA 141,11,21,24,169,0,109,9
63300 DATA 21,168,169,32,109,8,21,170
63310 DATA 152,109,7,21,168,138,109,6
63320 DATA 21,170,152,109,10,21,141,13
63330 DATA 21,138,105,0,141,12,21,172
63340 DATA 13,21,173,12,21,133,252,169
63350 DATA 0,133,251,177,251,141,14,21
63360 DATA 169,1,174,11,21,240,4,10
63370 DATA 202,208,252,45,14,21,240,21
63380 DATA 173,2,21,56,237,3,21,141
63390 DATA 15,21,169,1,174,15,21,240
63400 DATA 4,10,202,208,252,24,109,5
63410 DATA 21,141,5,21,173,3,21,24
63420 DATA 109,4,21,238,2,21,205,2
63430 DATA 21,48,3,76,82,21,173,5
63440 DATA 21,32,210,255,24,173,1,21
63450 DATA 105,1,141,1,21,173,0,21
63460 DATA 105,0,141,0,21,169,1,205
63470 DATA 0,21,208,7,169,64,205,1
63480 DATA 21,240,3,76,65,21,169,13
63490 DATA 32,210,255,173,3,21,24,105
63500 DATA 7,141,3,21,201,196,176,3
63510 DATA 76,57,21,201,203,240,8,169
63520 DATA 3,141,4,21,76,57,21,169
63530 DATA 15,32,210,255,169,13,32,210
63540 DATA 255,32,210,255,169,4,32,195
63550 DATA 255,32,204,255,96
63560 END
```

CAMPUS

LABORATORIO SO

LA TUA UNIVERSITA' DEL SOFTWARE

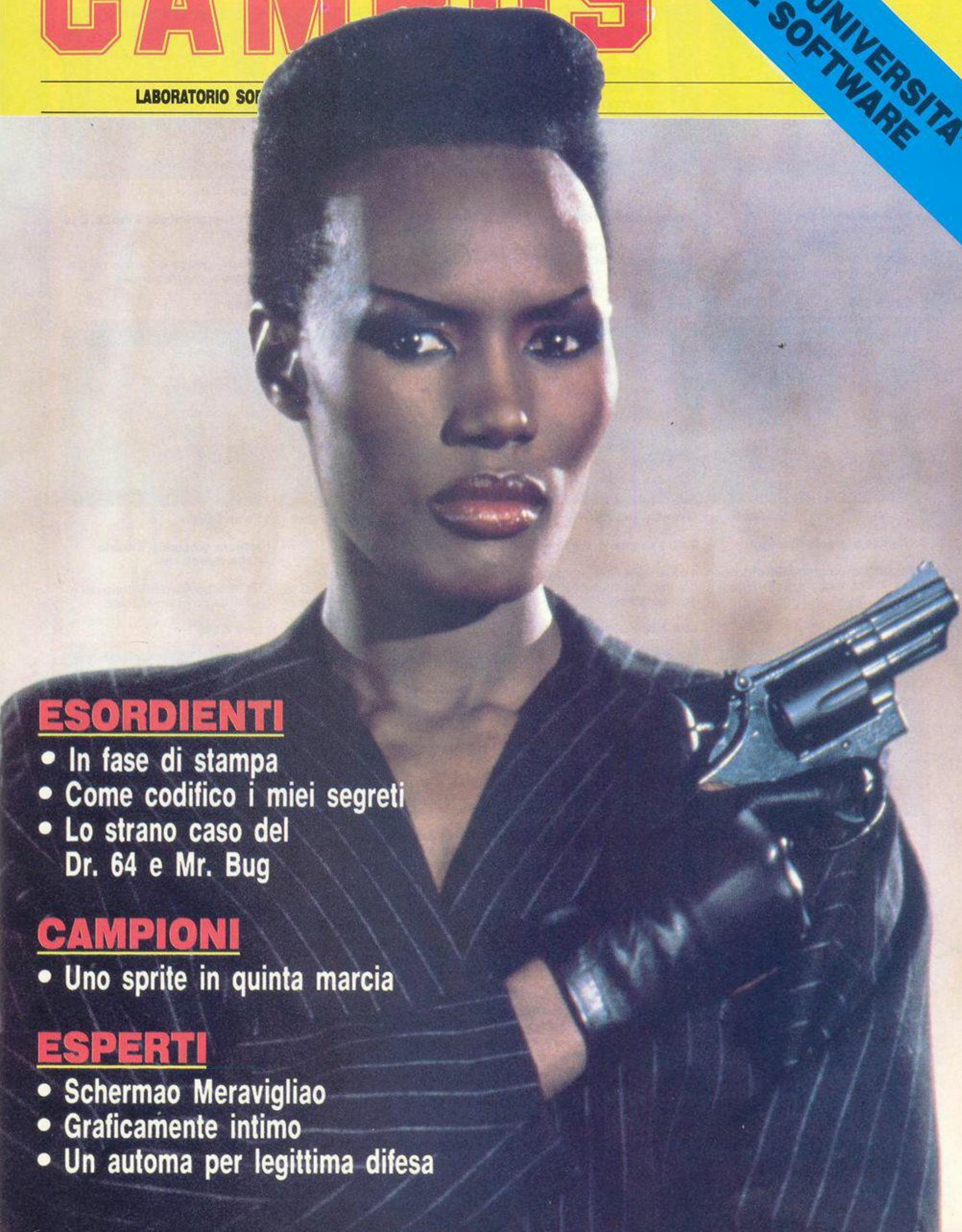
ESORDIENTI

- In fase di stampa
- Come codifico i miei segreti
- Lo strano caso del Dr. 64 e Mr. Bug

CAMPIONI

- Uno sprite in quinta marcia

ESPERTI

- Schermao Meravigliao
- Graficamente intimo
- Un automa per legittima difesa

La Grande Libreria Systems

Autori Vari

64 Programmi per Commodore 64

Giochi, grafica, gestione delle stringhe, musica, numeri, gestionali.

Lire 4.800

Autori Vari

I miei amici C16 & Plus4

Un manuale pratico per padroneggiare il basic di questi computer.

Lire 7.000

Autori Vari

Strategie vincenti per Commodore 64

Le strategie per tutti i classici del videogioco: per giocarli, vincerli o programmarli.

Lire 5.800

Autori Vari

62 Programmi per il Vic 20, C16 e Plus 4

Giochi, grafica e routine per imparare a programmare.

6.500

Roberto Didoni, Guido Grassi

Utilities e giochi didattici

Raccolta di programmi pratici per tutti i Commodore e lo Spectrum.

Lire 6.500

Giovanni Mellina

Tutti i segreti dello Spectrum

4 passi nella Rom: come usare le più importanti routine del sistema operativo.

Lire 7.000

Roberto Didoni, Guido Grassi

Simulazioni e test per la didattica

Teoria e listati per Vic 20, C16, C64 C128 e Spectrum Sinclair.

Lire 7.000

Paolo Goglio

Impara giocando il basic dello Spectrum

Esercizi pratici per entrare nel vivo della programmazione.

Lire 7.000

Clizio Merli

µPascal per Commodore 64/128

Un manuale completo per il programma compilatore

Lire 7.000

Umberto Colapicchioni e Luca Galuzzi

Dal registratore al drive del C64

Tutti i segreti delle memorie di massa del Commodore 64

Lire 7.000

Autori Vari

ADA

Il linguaggio passepartout dei computer degli anni '80.

Lire 5.000

Clizio Merli

Il linguaggio PASCAL

Un manuale tascabile per lo studio e la programmazione.

Lire 5.000

Sì, voglio arricchire la mia biblioteca con i seguenti volumi al prezzo di copertina + lire 3.000 per spese di spedizione.

- ☐ **64 Programmi per Commodore 64**
- ☐ **Strategie vincenti per i tuoi videogames**
- ☐ **62 Programmi per Vic 20 C16 e Plus77**
- ☐ **Utilities e giochi didattici**
- ☐ **Tutti i segreti dello Spectrum**
- ☐ **Simulazioni e test per la didattica**
- ☐ **Imparare giocando il basic dello Spectrum**
- ☐ **I miei amici C16 e Plus4**
- ☐ **Pascal per Commodore 128**
- ☐ **Dal registrattore al drive del C64**
- ☐ **ADA**
- ☐ **Il linguaggio Pascal**

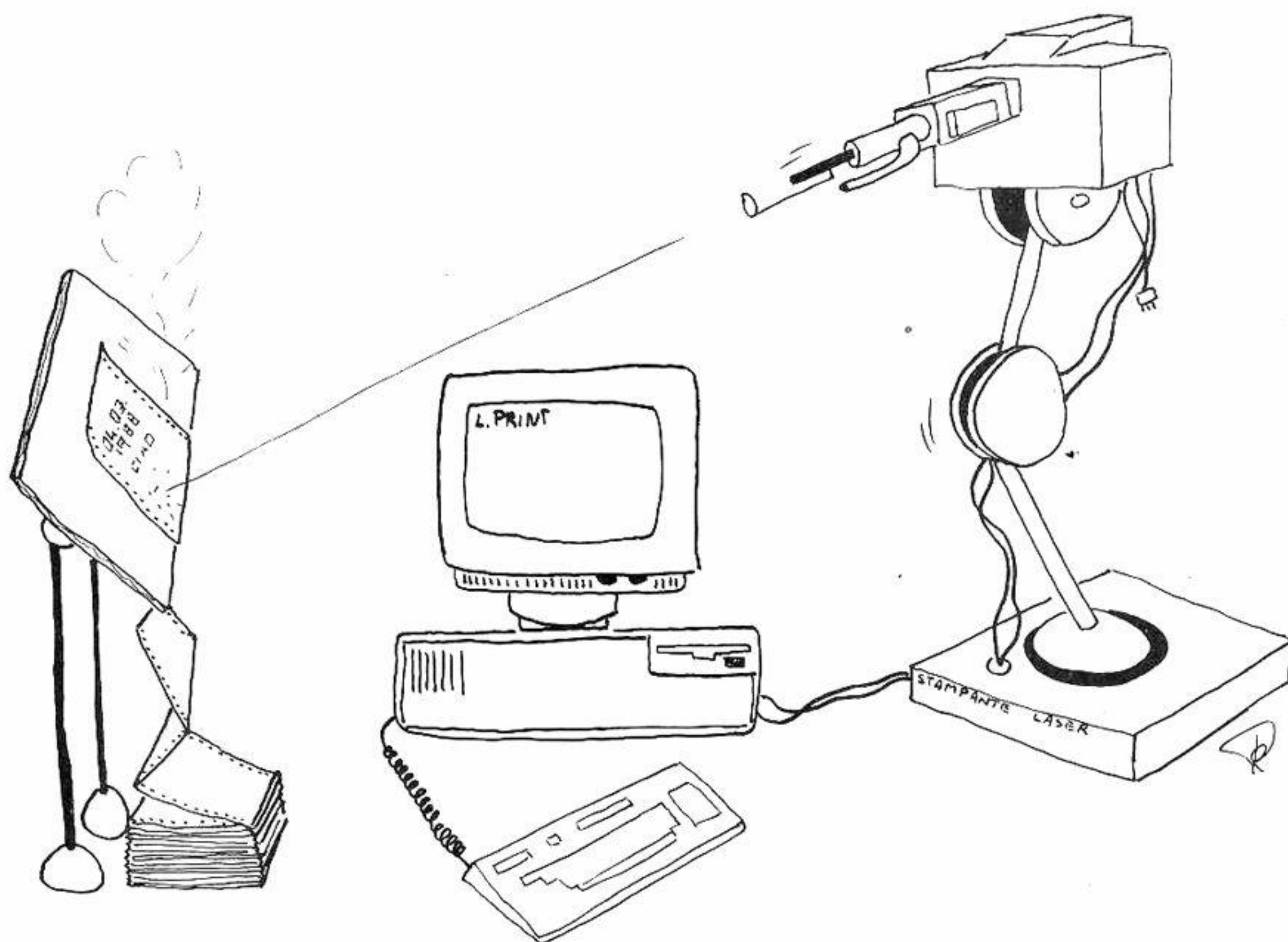
Nome ....

via .... **N.ro.** .... **telefono** ....

CAP .... **Città** ....

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Com puter* ☐ *VR Videoregistrare.* Pertanto vi invio la somma soltanto di lire ....

Valore dell'ordine lire....................................

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

IN FASE DI STAMPA

Esaminiamo una manciata di termini incontrati di frequente leggendo le caratteristiche delle stampanti

Spesso, sui cataloghi delle ditte specializzate, ci si imbatte in sigle sconosciute, vocaboli poco chiari, nomi inglesi di difficile traduzione.

Tale terminologia impedisce una scelta oculata dell'apparecchio che si intende acquistare; d'altra parte si rischia di procurarsi periferiche forse inadatte all'uso che si intende fare.

Cps

Questa sigla, relativa soprattutto alle stampanti, indica il numero di Caratteri Per Secondo che è possibile trasferire su carta.

Come si può intuire, maggiore è la velocità di scrittura, minore sarà il tempo impiegato a trascrivere un documento.

Le stampanti più veloci in assoluto sono quelle a tecnologia laser che, tra l'altro, offrono anche la migliore nitidezza di stampa.

Seguono le stampanti ad aghi; per ultime sono da ricordare quelle a margherita ed a sfera che permettono un'elevata qualità di stampa, ma al prezzo di una velocità piuttosto modesta.

Le caratteristiche di una stampante sono molto numerose

A volte non vale la pena acquistare una stampante "troppo" economica

Numero di aghi

Il numero degli aghi presenti in una testina di stampante (ovviamente del tipo ad aghi) è un indice validissimo della qualità offerta.

Le stampanti più economiche dispongono di 8 aghi; quelle "medie" offrono 9 aghi; le più care (e le migliori) arrivano a possedere 16 oppure 24 aghi, disposti su due file contigue.

Maggiore è il numero di aghi, migliore è la leggibilità delle righe stampate soprattutto per i caratteri che hanno le "gambette", meglio note come discendenti (cioè: q, y, g, j). Il maggior numero di aghi, di solito, consente la scrittura di esponenti e deponenti, impossibili da realizzare con stampanti economiche.

Bidirezionalità

Questa caratteristica non influisce sulla qualità di stampa, ma solo sulla velocità operativa.

Quando la testina di scrittura, stampando un rigo di testo, giunge al margine destro, deve tornare a sinistra facendo, nel contempo, avanzare il foglio di carta di un rigo.

Il tempo per tornare al margine sinistro, però, è un tempo "morto" che, nelle moderne stampanti, è utilizzato per stampare il rigo successivo, da destra a sinistra, ovviamente stampandolo al contrario, in modo da consentirne la successiva lettura.

Poichè la bidirezionalità è un problema facilissimo da risolvere a livello costruttivo, al giorno d'oggi quasi tutte le stampanti prodotte sono bidirezionali; quelle monodirezionali sono di vecchia produzione oppure super-extra-economiche.

In modo grafico, tuttavia, quasi tutte le stampanti sono monodirezionali.

NLQ

Questo termine significa Near Letter Qauality (qualità di stampa simile a quella di una macchina da scrivere).

Il martelletto di una vetusta macchina da scrivere offriva, fino a qualche anno fa, qualità e nitidezza impensabili per una stampante ad aghi. Quest'ultima, infatti, stampa il carattere grazie alla battitura su carta di un gruppo di puntini (dot) affiancati, che riproducono, più o meno fedelmente, il carattere alfanumerico.

Con le moderne stampanti ad aghi il problema è stato risolto facendo stampare uno stesso rigo due volte, sfalsato, però, di una frazione di millimetro in modo che gli aghi, battendo lo stesso carattere (sfalsato) per la seconda volta, "coprono" le parti bianche intermedie tra un puntino e quello sottostante.

I caratteri stampati hanno così ben poco da invidiare a quelli offerto dai martelletti di antica memoria, ma la velocità di stampa scende in maniera impressionante: da 120 cps, ad esempio, si può arrivare a soli 30 cps.

Buffer

Quando un computer invia dati da stampare, di solito non può essere impiegato per altre elaborazioni perchè, appunto, è occupato ad inviar dati alla periferica.

Se, ad esempio, il file da inviare alla stampante è molto lungo (come nel caso di testi trattati con un word processor), il tempo impiegato può arrivare a parecchie decine di secondi. Tale tempo è necessario perchè la stampante è più lenta del computer e, di conseguenza, se ha ricevuto un rigo da stampare invia un segnale al computer per impedire l'invio del rigo successivo: quando, la stampa del rigo termina, la periferica chiede l'invio di un altro rigo, e così via fino alla fine.

Molte stampanti dispongono di un buffer, vale a dire di un banco di memoria Ram che, come tale, è in grado di ricevere molti dati dal computer a velocità elettronica.

Se il buffer è di 16 KRam (vale a dire oltre 16000 byte) sarà possibile inviare alla stampante un documento di tale lunghezza in un battibaleno; subito dopo potremo continuare a lavorare con l'elaboratore, mentre la stampante provvederà a stampare il file memorizzato nel buffer.

Matrice del carattere

Un carattere alfanumerico, qualunque esso sia, è "racchiuso" all'interno di una matrice rettangolare di puntini (dot) che, anneriti o lasciati bianchi, consentono di rendere intellegibile lo stesso carattere.

E' ovvio che maggiore è il numero di punti (cioè la dimensione della matrice) più nitido sarà il carattere.

Le stampanti più economiche hanno una matrice di 7x7 caratteri; le migliori arrivano a 24x31.

Formato

La dimensione massima del foglio adoperabile è importante per chi desideri utilizzare particolari pacchetti applicativi che consentono di scrivere su un foglio protocollo aperto.

Di solito la possibilità di adoperare un formato A4 (21 x 29.6 cm.) è più che sufficiente per l'hobbysta, anche se "evoluto". Quasi sempre, inoltre, le stampanti migliori offrono la possibilità di adoperare il formato maggiore (A3).

Interfacce

Non tutte le stampanti sono collegabili a tutti i computer.

Nel caso particolare del C/64, C/128, C/16 e Plus/4, una stampante deve essere Commodore compatibile altrimenti il collegamento non è assolutamente possibile.

Di solito una stampante viene venduta con almeno un'interfaccia incorporata.

Nel campo professionale lo sytandard più diffuso è il Centronics, ma è possibile trovare stampanti dotate di protocollo Rs-232, IEEE-488 oppure dotate di software particolare in grado di emulare una o più interfacce.

Font

Il "tipo" di carattere è quasi sempre uno solo nelle stampanti economiche; spesso sono due e, sulle stampanti migliori, sono disponibili molti tipi diversi, selezionabili anche inserendo particolari schede che contengono set di caratteri alternativi.

Indubbiamente una lettera, ed un documento in generale, presenta un aspetto decisamente migliore se alcuni brani vengono riprodotti con font diversi.

Trazione

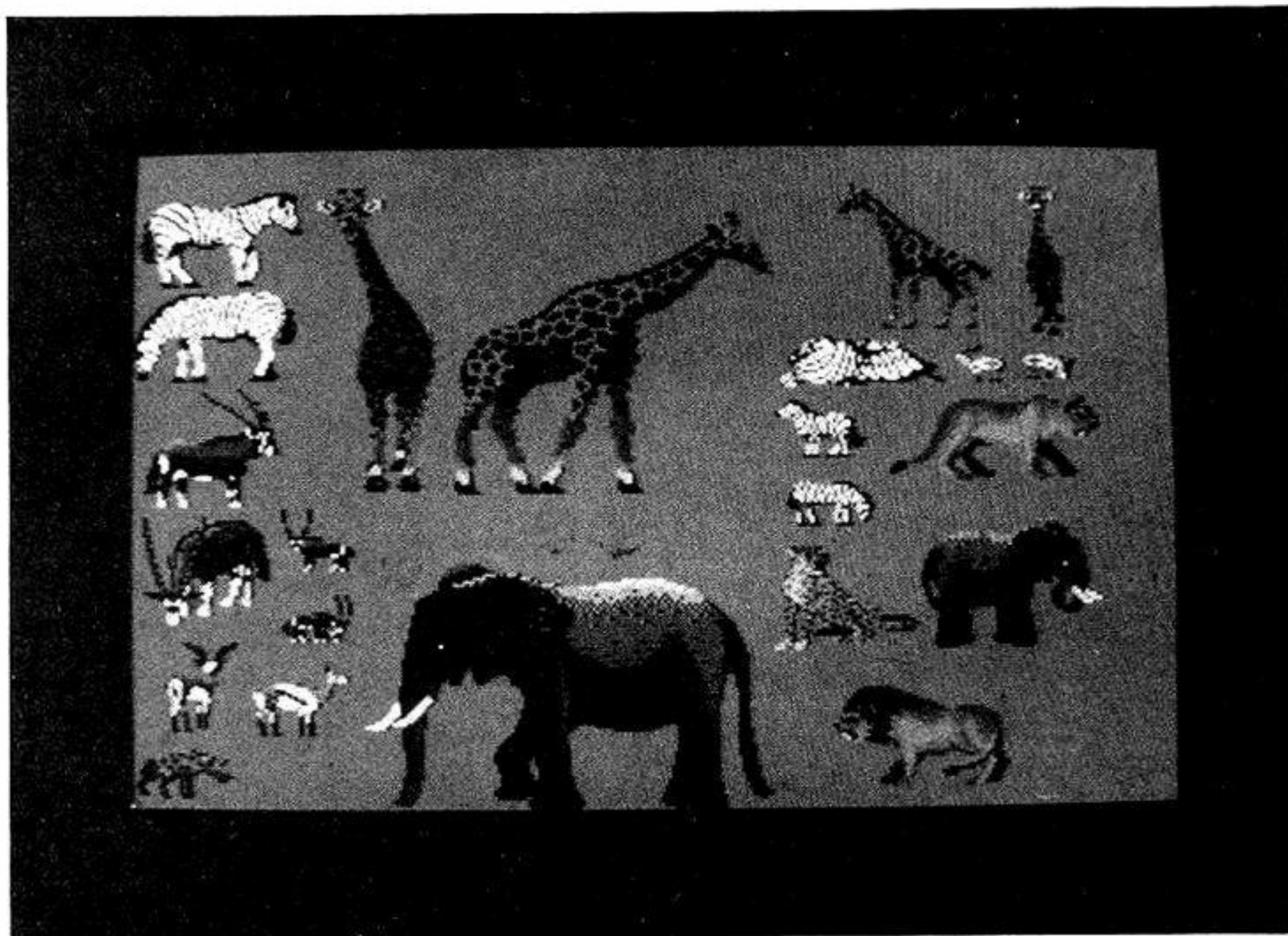
Una stampante, per essere considerata realmente versatile, deve avere la capacità di utilizzare sia fogli singoli che in modulo continuo (quelli, cioè, dotati di forellini ai due lati destro e sinistro).

Deve quindi disporre di un sistema a frizione (per i fogli singoli) e di un "trattore", vale a dire di un cilindro, dentato alle due estremità, che possa "addentare" i bordi forati del modulo continuo, per favorirne l'avanzamento.

Per una qualità di stampa impeccabile è spesso necessario rinunciare ad elevate velocità

Dizionario dell'informatica

E' disponibile, presso il nostro servizio arretrati, il dizionario Inglese - Italiano dell'informatica dedicato ai lettori che s'avvicinano per la prima volta all'affascinante mondo dell'informatica, agli specialisti che hanno l'esigenza di accedere alla dinamica letteratura anglosassone ed a quanti, insomma, nelle aziende produttrici di macchine e di software hanno il non facile compito di preparare materiale didattico.

Per informazioni telefonare a: 02/84.67.34.8

UN APPROCCIO GRAFICAMENTE INTIMO

Cerchiamo di sapere qualcosa di più sulla gestione del sistema grafico di Amiga

di **Luigi Callegari**

La grafica è il fiore all'occhiello dell'Amiga

La libreria grafica

La libreria grafica di supporto, presente nella ROM di Amiga, ha due scopi fondamentali: semplificare l'interfaccia con il complesso meccanismo hardware di visualizzazione usato in Amiga e fornire procedure di gestione dei chip grafici.

Dal momento che ci siamo prefissi di illustrare solo a grandi linee la ROM di Amiga (lo spazio è tiranno) parleremo solo delle funzioni basilari grafiche disponibili, in modo che almeno si sappia che cosa è già stato fatto dagli autori della ROM.

La libreria grafica fornisce parecchie strutture di dati per definire e manipolare la memoria grafica: ricordiamo che il sistema è stato scritto in buona parte in BCPL e C, quindi l'ideale è gestirlo proprio usando questi linguaggi.

La memoria grafica è posta in un banco di RAM, organizzata in "piani" di informazioni sui pixel.

Si badi bene, dunque, che non è la sòlita "Bit-Mapped graphics" simile a quella di Apple, Spectrum o C/64. Qui, infatti, ogni piano, owero ogni pagina di locazioni consecutive di memoria, contiene un solo bit di

informazione per ogni punto (pixel) sul video.

Negli attuali esemplari di Amiga (500 - 1000 - 2000) per creare un video vengono usati, di solito, da 1 a 5 piani (chiamati, nella documentazione tecnica, "Bitplanes"). Ogni pixel sullo schermo, dunque, può essere controllato INDIRETTAMENTE, come vedremo tra breve, da uno sino a cinque bit, ovvero da un numero binario compreso tra 0 e 31 (cinque piani).

"Indirettamente" perchè questo valore, in realtà, specifica uno dei 32 registri hardware di colore di Amiga contenuti nel chip grafico. Ogni registro, negli attuali Amiga, è "largo" 12 bit, sicchè ogni registro può contenere 4096 gradazioni di colore, tutte diverse e non "sfumature", come in Apple II-GS. Il valore a 12 bit contenuto in un registro di colore è suddivisibile in tre porzioni di quattro bit ciascuna, indicanti l'intensità (da 0 a 15) di uno dei tre colori fondamentali: rosso, verde e blu che, mescolati tara loro, permettono di ottenere tutti i colori dello spettro ottico.

Le due strutture fondamentali di dati presenti nella libreria grafica sono: BitMap e RastPort. La prima definisce quanti piani di bit si vogliono usare. La struttura denominata RastPort, invece, è un insieme di parametri destinati ad essere trasferiti nella struttura definita da BitMap.

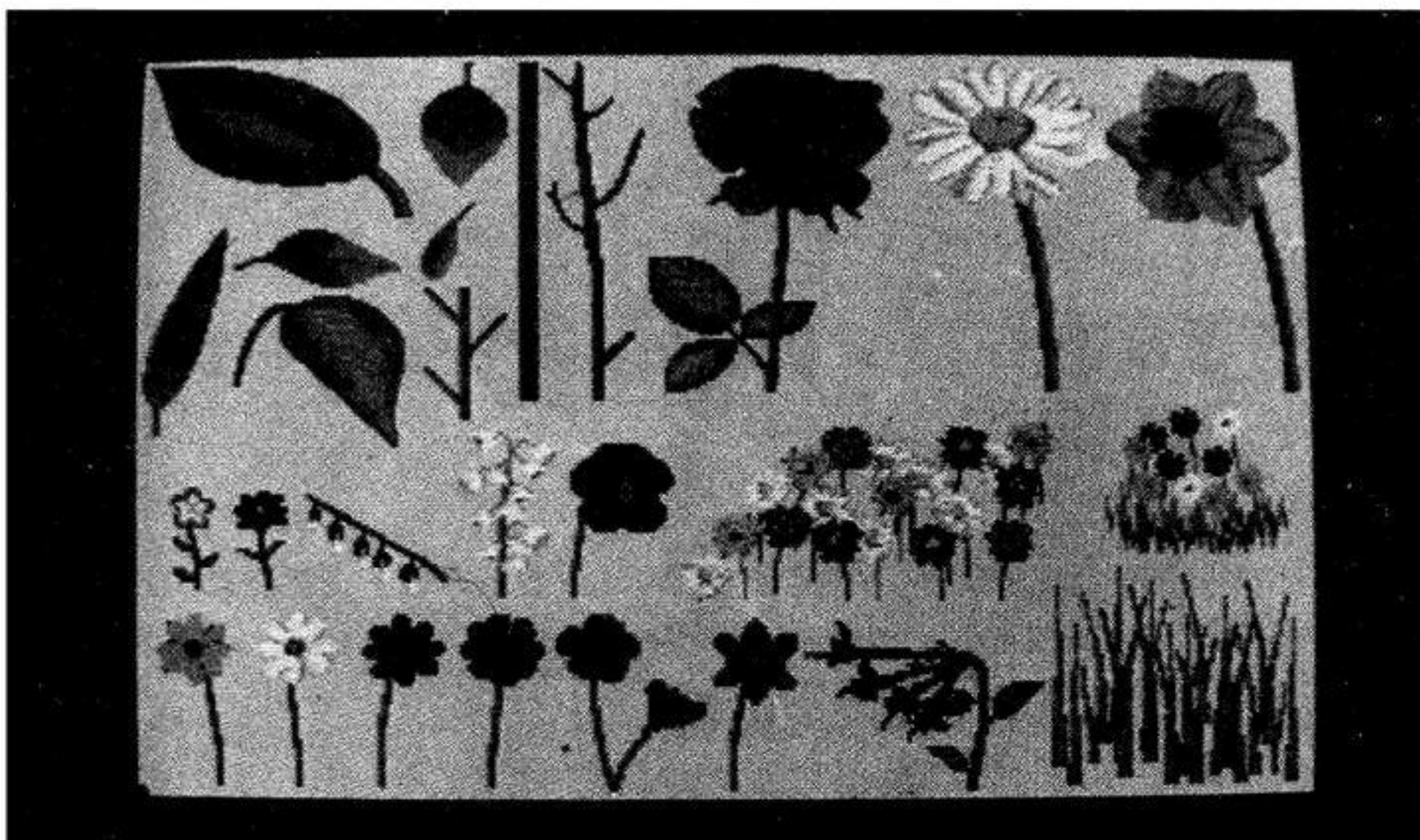
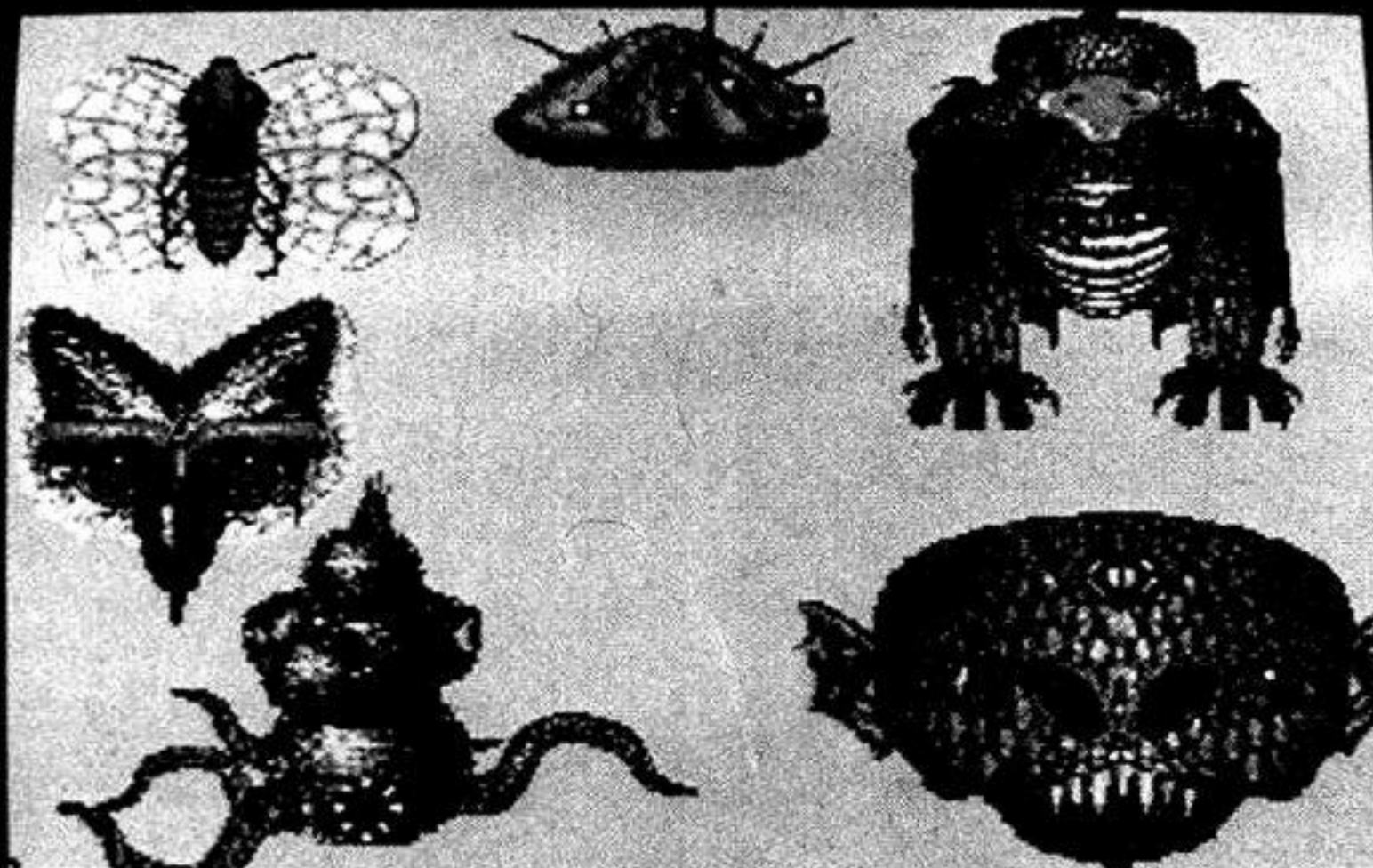
Questi parametri vengono fissati dalle routines grafiche, dalle routines di testo e dagli oggetti animati.

Vediamo un elenco dei componenti principali di una struttura RastPort:

1) Penna principale. FgPen (ForeGroundPen) è la "penna" principale di tracciatura. Quando si usa una sola penna (tracciatura linee e riempimenti di aree) questa è quella usata dal sistema.
2) Penna secondaria. BgPen (BackGround Pen). Si tratta della penna usata quando ne sono necessarie due, ad esempio durante la scrittura di testi, ove vi è un testo (foreground) ed uno sfondo (background) diversi per colore.
3) Modo di tracciatura (DRAW MODE). Specifica come devono essere visualizzati i dati inviati al BitMap. Con effetti differenti a seconda che si tratti di grafica, riempimenti o stampa di testo, i modi principali sono: JAM1, JAM2, COMPLEMENT ed INVERSEVID, parole familiari a chi usa l'editor di icone di Workbench.

Le penne grafiche

Quando la libreria grafica esegue l'interpretazione di dati, usa la struttura RastPort che descrive sia il registro di colore, sia la posizione di visualizzazione usata nella tracciatura del testo.

La routine SetApen() fissa il valore della penna usata come numero di registro colore da usare per materializzare la sua "scia". Le coordinate della penna possono essere spostate usando Move().

Linee, rettangoli e riempimenti

Tracciare una linea (in C) richiede solo quattro passi: si fissa dapprima la penna (con le limitazioni dovute al numero di Bitplanes scelti); poi si sceglie il modo grafico: JAM1 è il più tipico. Poi si sposta la penna nella posizione voluta e si traccia la linea. Una procedura C generalizzata potrebbe essere simile a quella che compare in queste pagine.

Si noti che i nomi delle funzioni sono tutte presenti nella libreria grafica in ROM e quindi anche nelle librerie dei compilatori C per Amiga: Move(), Draw(), SetAPen(), Set DrMd().

Si noti che Move() sposta la penna, mentre Draw() la sposta tracciando una linea del colore selezionato (come in LOGO). La funzione RectFill() usa la penna principale per riempire un'area rettangolare. Usando il blitter, quest'ultima routine può creare un centinaio di quadrati di 96 x 96 pixel di lato in un bitplane ogni secondo.

Testo

La penna grafica è usata come origine della stampa anche dei testi. La sua posizione, cioè, indica dove andrà scritto il primo carattere. La coordinata X specifica la posizione del primo carattere, la coordinata Y descrive la linea dove "appoggiano" i caratteri alfabetici, mentre i discendenti scendono sotto di essa.

Ovviamente le funzioni SetAPen(), Set BPen() e SetDrMd() sono usate anche per i testi, viste le premesse, mentre la funzione Text() per scrivere testi sul video, senza preoccuparsi di aggiornare la posizione, dato che dopo ogni chiamata la posizione nuova del cursore grafico viene mantenuta in memoria.

Uno sguardo ad Intuition

Intuition è frutto dell'adattamento del sistema operativo "ad icone" originalmente applicato per la prima volta dalla Apple nel computer "LISA" per semplificare la vita ai non addetti. In Amiga, Intuition è anche uno strumento praticamente insostituibile di programmazione, fedele compagno del suo utilizzatore ed in grado di alleviargli molte fatiche.

L'unità base di INTUITION è lo schermo ("screen"). Questo è formato da una mistura di dati e di istruzioni per lo hardware di A-

```
DrawLine (RPort, Pen, Mode, StartX, StartY, EndX, EndY)
    {
     SetAPen(RPort, Pen);
     SetDrMd(RPort, Mode);
     Move(RPort, StartX, StartY);
     Draw(RPort, EndX, EndY);
    }
```

miga. Per creare uno schermo si usa la libreria grafica. Eventualmente si può fare ciò senza usare Intuition, ma risulta molto più difficile. Non per niente Intuition è stato concepito proprio per facilitare l'interfaccia col sofisticato hardware interno; tutto può essere svolto semplicemente con due passi: definire una struttura chiamata NewScreen (ideale è, come al solito, programmare in C) e usando la funzione OpenScreen() di Intuition.

Per mezzo della definizione di NewScreen si può fissare semplicemente le dimensioni della finestra, la sua risoluzione (alta, bassa, interlacciata) eccetera. Inoltre Intuition permette la coesistenza di più finestre attive in contemporanea, ciascuna controllata dal suo task.

Finestre

Le finestre sono praticamente dei piccoli schermi posizionabili all'interno degli schermi propriamente detti. Possono essere riposizionate, ridimensionate, traslate, chiuse, via mouse, direttamente da Intuition. Una window è un vero e proprio terminale agli "occhi" del sistema, tanto da rispondere rigidamente alle norme ANSI di comunicazione. Perciò un programma può essere scritto per inviare il suo output ad una finestra esattamente come se fosse un terminale remoto, con grande vantaggio in termini di "portabilità" su altre macchine.

Le finestre possono essere dotate di "gadgets" definiti dall'utente che, selezionati da mouse, possono attivare specifiche funzioni ed altri task direttamente da Intuition.

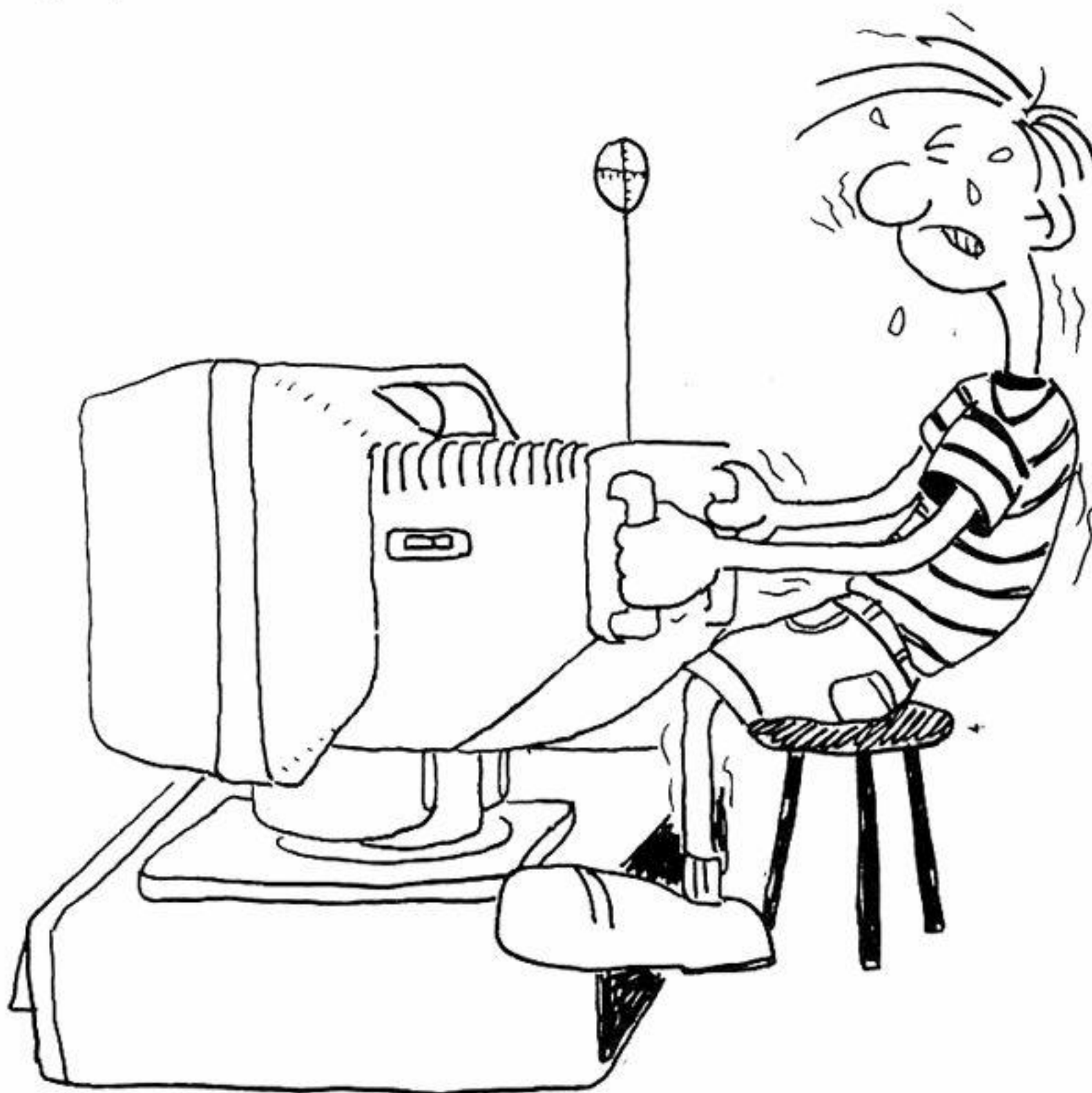
Nelle finestre si può facilmente scrivere usando set di caratteri diversi dal classico "topaz".

Menu

Anche il classico sistema di "menu a sipario", gestiti da mouse, è implementato da Intuition. Esso permette di semplificare e potenziare l'interfaccia del programma verso l'utente, che in ogni momento può richiedere l'attivazione di un task o selezionare una procedura. Le opzioni possono essere agevolmente indicate anche tramite pressioni di combinazioni di tasti, sempre gestiti da Intuition in comunione con Exec.

Intuition controlla il programma da menu sfruttando le porte di comunicazione speciali, chiamate IDCMP (Intuition Direct Comunications Message Ports), ovvero porte per la comunicazione diretta di messaggi di Intuition.

I messaggi possono essere trasmessi dalla tastiera, da altri task, dal mouse, dall'interfaccia disco e da tutte le periferiche di sistema.

Una finestra è un vero e proprio "terminale" grafico

UN AUTO-RUN CHE AGISCE PER LEGITTIMA DIFESA

"Tentar non nuoce" dice il proverbio; e tentare di proteggere i propri listati porta, se non altro, ad interessanti scoperte sul modo di agire del vostro C/64

di **Domenico Pavone**

L'applicazione di una protezione consente di scoprire nuove procedure di programmazione

Il listato di queste pagine, che consente di applicare la protezione, è facilmente comprensibile grazie alla gran mole di REM esplicative che potete omettere di trascrivere se temete di stancare le vostre dita.

Prima di tutto è richiesto, in fase di Input, il nome del programma che si vuole dotare di autostart; come potete notare dalla linea 40 (e successivamente nella linea 65) per evitare la visualizzazione del punto interrogativo si è usata la tecnica di aprire un file in lettura dalla tastiera (periferica n.0).

Al nome viene poi dato, nella variabile PO$, il suffisso "/auto", che lo tronca se supera gli undici caratteri.

Occorre poi dare il testo del messaggio che, per motivi estetici, è stato limitato a 36 caratteri, compresi eventuali spazi. Poichè (come vedremo più avanti) il testo dovrà essere inviato al disco, carattere per carattere, la linea 70 crea un vettore di 36 elementi ognuno dei quali contiene il valore di schermo del carattere corrispondente (in pratica quello che dovreste trascrivere nell'area di memoria da 1024 a 2023 per vederlo apparire) sommato a 128, per ottenere l'effetto "reverse".

A questo punto viene chiesto se si vuole, o meno, la protezione antireset e, prima di cominciare l'elaborazione vera e propria, la riga 85 disabilita lo schermo per velocizzare, an-

che se di poco, l'esecuzione del programma.

Comincia ora la vera e propria routine di creazione dell'autostart, che può essere schematizzata nelle seguenti operazioni:

• Apre in lettura il file programma da trattare e ne controlla i primi due byte, che contengono l'indirizzo dal quale lo stesso viene caricato in memoria nel solito formato basso/alto.

Poichè Run & Message può funzionare solo su programmi che iniziano dalla locazione 2049 (quindi scritti in Basic o in linguaggio macchina, purchè contengano una riga del tipo: 10 sys...), i due bytes letti dovranno essere uguali ad 1 il primo e a 8 il secondo (8*256+1=2049); in caso contrario una segnalazione di errore arresta il programma (righe 105-120).

• Apre in scrittura un file programma con il suffisso "/AUTO" e ne registra i due bytes che indicheranno l'indirizzo dal quale il file verrà caricato.

Questo indirizzo dovrà essere 719 o 758, a seconda se si sia scelto di includere, o meno, l'antireset; quindi il byte basso da caricare sarà 207 o 246, mentre il byte alto sarà 2 (righe 135-145).

• Nel primo caso verrà letta dalle linee DATA una routine LM di antireset e trascritta, byte per byte, sul file aperto in scrittura su disco (righe 155-165).

Questa routine è molto semplice (vedi anche C.C.C. n.30) e si limita ad abbassare il tetto del basic per proteggere le locazioni da 32768 in poi; quindi alloca l'indirizzo della routine di autorun e trasferisce i caratteri "CBM80" che sono già presenti nella memoria a partire da 64738 (monitorare per credere) per simulare la presenza di una cartuccia esterna.

Se si è scelto di non avere l'antireset, verrà trascritta, sempre subito dopo i due byte che ne indicano l'indirizzo di caricamento, solo la routine di autorun (riga 175), che è tratta dal n.4 della compianta rivista "Commodore".

• Di seguito alle routine sopradette, che quando saranno caricate occuperanno le locazioni fino a 767, dovrà ora essere accodata (sempre nel file ".../AUTO") l'area di memoria compresa tra 768 e 819, che comprende tutta una serie di vettori che puntano a varie routine

La disabilitazione del tasto Run-Stop è una premessa indispensabile per qualsiasi protezione

PROTEGGERE E' UN PO' IMPARARE

Non sempre le protezioni svolgono efficacemente il proprio lavoro; soprattutto da quando circolano cartucce ed utility sprotettrici di cui noi stessi ci occupiamo in queste pagine.

Tuttavia la pubblicazioni di articoli di questo genere non hanno il solo scopo di rendere disponibile una procedura ma, soprattutto, di invogliare l'utente quasi-principiante a soffermarsi su routine in linguaggio macchina, magari da comprendere e modificare in modo da inserirle nei propri package.

Senza considerare la soddisfazione di realizzare, con l'aiuto di alcuni codici macchina, qualcosa che non tutti sanno!

Dopo la più che esauriente serie di articoli (apparsi sin dal n.4 della rivista "Commodore" e poi proseguiti su C.C.C. a partire dal numero 27), infatti, può sembrare inutile tornare su un argomento come l'autorun, ma pur non inventando niente di totalmente nuovo c'è un altro modo di accostarsi al tema, ed è quello della sintesi e della semplificazione di quanto già noto.

E' quello che si propone il programma "Run & Message" che, oltre a fornire di autostart un programma Basic su disco, provvede in maniera molto semplice, e usando soprattutto il buon vecchio Basic, a visualizzare, durante il caricamento del programma, un messaggio di vostra scelta, in reverse e centrato sullo schermo.

Run & Message inibisce automaticamente l'azione dei tasti Run/Stop e Restore e consente, inoltre, di selezionare l'inserimento, o meno, di una protezione antireset; consente addirittura, se ne siete in grado, di introdurre altri tipi di protezione modificando alcuni DATA ben evidenziati nel listato.

Il file-programma generato dalla procedura, ovviamente, andrà caricato nella forma sintattica ",8,1".

Unico neo del programma può essere considerata la sua relativa lentezza di esecuzione dovuta alla notoria "calma" con cui si svolgono le operazioni di input/output verso la periferica, ma in fondo il trattamento di autorun va fatto una sola volta.

```
                DISASSEMBLATO ROUTINE DI ANTIRESET

02CF    LDA #$FF       Carica in accumulatore 255 (decimale).
02D1    STA $33        Mette 255 in locazione 51
02D3    STA $37        e in locazione 55.
02D5    LDA #$7F       Carica in accunulatore 127 (dec).
02D7    STA $34        Mette 127 in locazione 52
02D9    STA $38        e in locazione 56.
02DB    LDA #$F6       Carica in acc. 246.
02DD    STA $8000      Mette 246 in locazione 32768
02E0    STA $8002      e in locazione 32770.
02E3    LDA #$02       Carica in acc. 2.
02E5    STA $8001      Mette 2 in locazione 32769
02E8    STA $8003      e in locazione 32771.
02EB    LDX #$05       Carica 5 nel registro X.
02ED    LDA $FD0F,X    Carica in acc. il contenuto
                       della locazione 64738+X.
02F0    STA $8003,X    Mette il contenuto dell'accumulatore
                       in locazione 32771+X.
02F3    DEX            Decrementa di uno il valore di X.
02F4    BNE $02ED      Se X <> 0 salta a 02ed,
                       altrimenti prosegue.


                DISASSEMBLATO ROUTINE DI AUTOSTART

02F6    JSR $A656      Effettua un CLR.
02F9    JSR $A68E      Fa puntare l'interprete
                       all'inizio del programma.
02FC    JMP $A7AE      Esegue l'istruzione che segue.
```

di sistema.

Questi sono inseriti nelle linee DATA da 440 a 510, ad eccezione del warm start che è trascritto dalla linea 190. Le REM indicano quelli più interessanti al fine di eventuali protezioni, ed indicano (tra parentesi) i valori che normalmente hanno quelli modificati (per esempio il 234 della riga 495 che blocca il tasto Run/Stop).

Ovviamente potete sbizzarrirvi nel modificare le combinazioni che più ritenete utili, tenendo presente che i valori che puntano alla routine di Run sono: 246,2.

• A questo punto c'è ancora da riversare nel file l'area che va da 820 a 2047, che comprende il buffer di cassetta e la memoria di schermo.

Queste locazioni sono riempite con spazi (CHR$32), fatta eccezione per quelle che identificano tre linee centrali dell'area di schermo: queste conterranno spazi in reverse (CHR$160), al cui interno verranno visualizzati i caratteri che compongono il messaggio (linee 205-230).

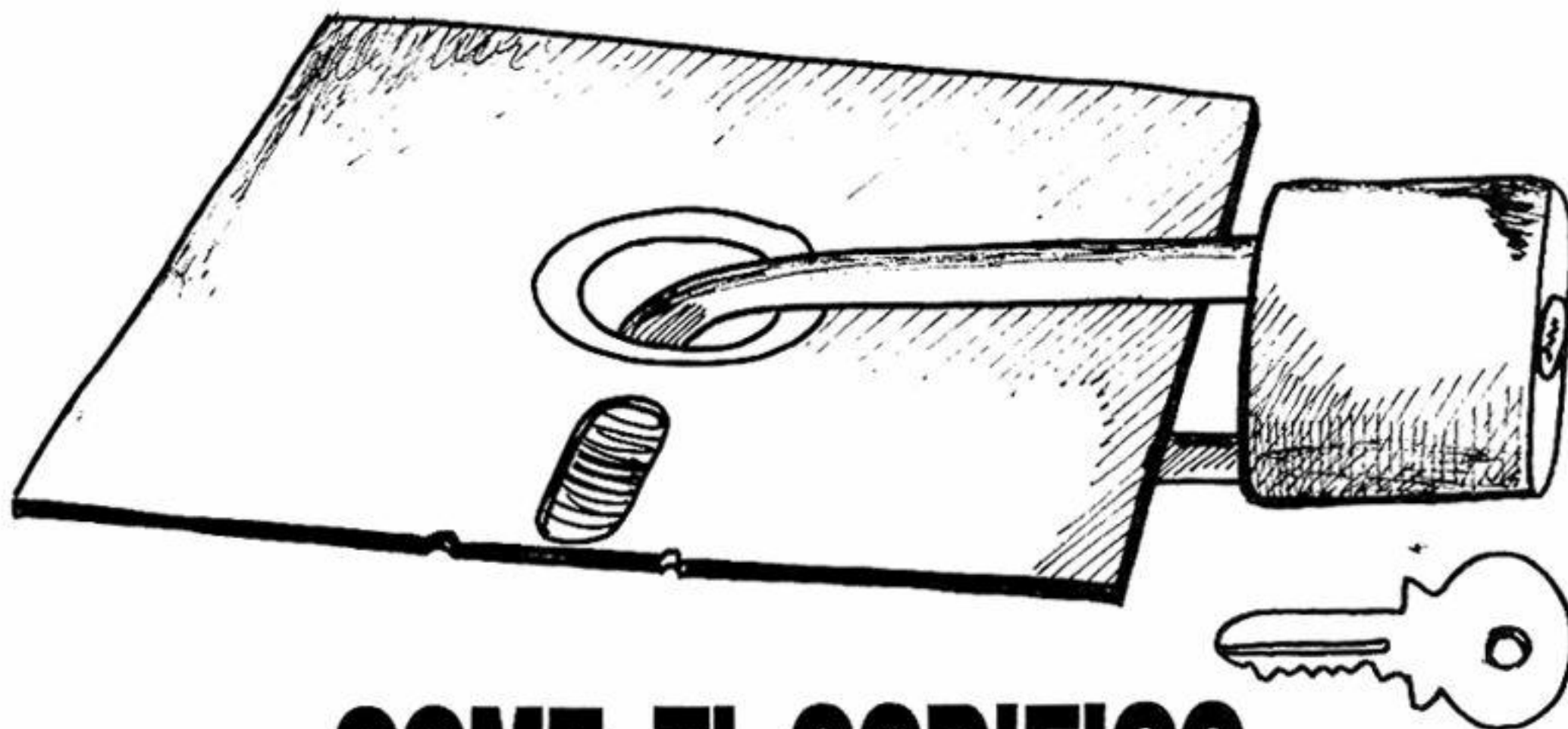
• Non rimane altro, ora, che mettere uno zero (come d'obbligo) alla locazione 2048, poi copiare in coda a quanto già fatto il programma originale leggendolo byte per byte, dal file aperto in lettura, e trascrivendolo in quello in scrittura (linee 250-255).

Un'ultima osservazione prima di concludere: fate attenzione a modificare il vettore della routine di errore o di break; accertatevi che il programma da trattare sia veramente corretto, perchè una volta elaborato non potrà più indicarvi l'ubicazione di un eventuale errore.

Un altro consiglio è quello di tener presente che se il programma si conclude con un semplice END, una volta dotato di autostart, invece di fermarsi col solito READY ripartirà immediatamente!

Se volete, quindi, un ritorno alle condizioni normali, il programma dovrà provvedere a ripristinare i valori di default di warm start.

```
5 PRINTCHR$(147):POKE 53280,0
  :POKE 53281,0:PRINTCHR$(155
  ):DIM  MS(36)
10 P$="PROGRAMMA ":DT$="DA TRA
   TTARE?":NT$="NON TRATTABILE
   "
15 FOR I=1 TO 36:TR$=TR$+"-":N
   EXT:UP$=CHR$(145)+CHR$(145)
20 REM ----------------------
   ----------
25 REM  * INPUT NOME PROGR. E
   MESSAGGIO
30 REM ----------------------
   -----------
35 PRINTCHR$(147)SPC(204)SPC(2
   04)P$+DT$:PRINT:PRINT
40 PRINTSPC(11)LEFT$(TR$,16)UP
   $:PRINTSPC(11);:OPEN 1,0:IN
   PUT#1,PS$:CLOSE 1
45 IF LEN(PS$)>16 THEN 35
50 PO$=LEFT$(PS$,11)+"/AUTO":P
   RINTCHR$(147);
55 PRINTSPC(227)SPC(228)"MESSA
   GGIO?":PRINT:PRINTSPC(11)"(
   MAX 36 CARATTERI)"
60 PRINT:PRINT:PRINTCHR$(32)CH
   R$(32)TR$+UP$:PRINTSPC(2);
65 OPEN 1,0:INPUT#1,MS$:CLOSE
   1:Z=LEN(MS$):IF Z>36 THEN 5
   5
70 FOR I=1 TO Z:MS(I)=PEEK(162
   5+I)+128:NEXT
75 PRINTCHR$(147)SPC(220)SPC(2
   26);:INPUT "VUOI L'ANTIRESE
   T?  (S/N)";W$
80 IF W$<>"S" AND W$<>"N" THEN
    75
85 POKE 53265,PEEK(53265) AND
   239:REM  * DISABILITA SCHER
   MO *
90 REM ----------------------
   ----------
95 REM  ****  OPERAZIONI SU DI
   SCO  ****
100 REM ----------------------
    ----------
105 OPEN 15,8,15:OPEN 5,8,5,PS$
    :GOSUB 335
110 GET #5,A$,B$
115 REM  -- CONTROLLA SE IL PRO
    GRAMMA COMINCIA DA 2049
120 IF A$<>CHR$(1) OR B$<>CHR$(
    8) THEN PRINTCHR$(147)SPC(2
    47)P$+NT$:GOTO 280
125 OPEN 6,8,6,PO$+",P,W":GOSUB
     335
130 REM  --- INDIRIZZO DI CARIC
    AMENTO ---
135 IF W$="S" THEN W=207
140 IF W$="N" THEN W=246
145 PRINT#6,CHR$(W);:PRINT#6,CH
    R$(2);
150 REM  -- LEGGE E INSTALLA AN
    TIRESET --
155 FOR X=1 TO 39:READ Y:IF W$=
    "N" THEN 165
160 PRINT#6,CHR$(Y);
165 NEXT
170 REM  --- LEGGE E INSTALLA A
    UTORUN ---
175 FOR X=1 TO 10:READ Y:PRINT#
    6,CHR$(Y);:NEXT
180 REM  -- LEGGE E INSTALLA VE
    TTORI ----
185 FOR X=768 TO 769:READ Y:PRI
    NT#6,CHR$(Y);:NEXT
190 PRINT#6,CHR$(W);:PRINT#6,CH
    R$(2);:REM  ** WARM START (
    DEFAULT 131,164)
195 FOR I=772 TO 819:READ Y:PRI
    NT#6,CHR$(Y);:NEXT
200 REM  ---  INSTALLA MESSAGGI
    O E RIMANENTE AREA DI MEMOR
    IA FINO A 2047  -------
205 FOR X=820 TO 1463:PRINT#6,C
    HR$(32);:NEXT
210 CI=1503+INT((40-Z)/2)
```

```
215 FOR X=1464 TO CI:PRINT#6,CH
    R$(160);:NEXT
220 FOR X=CI+1 TO CI+Z:J=J+1:PR
    INT#6,CHR$(MS(J));:NEXT
225 FOR X=CI+Z+1 TO 1583:PRINT#
    6,CHR$(160);:NEXT
230 FOR X=1584 TO 2047:PRINT#6,
    CHR$(32);:NEXT
235 REM  - METTE ZERO ALLA LOCA
    ZIONE 2048
240 PRINT#6,CHR$(0);
245 REM  ----  LEGGE IL PROGRAM
    MA E LO ACCODA NEL NUOVO FI
    LE  -------------------
250 GET #5,A$:S=ST:IF LEN(A$)=0
     THEN A$=CHR$(0)
255 PRINT#6,A$;:IF S=0 THEN 250
260 REM -----------------------
    ----------
265 REM  ****** ROUTINE DI FINE
     *******
270 REM -----------------------
    ----------
275 PRINTCHR$(147)SPC(147)SPC(1
    47)"F A T T O"
280 CLOSE 5:CLOSE 6:CLOSE 15
285 POKE 53265,PEEK(53265) OR 1
    6: REM  * RIABILITA SCHERMO
     *
290 PRINT:PRINT:PRINT:PRINT:PRI
    NTSPC(8)"< F1 > - PER CONTI
    NUARE":PRINT
295 PRINTSPC(8)"< F7 > - PER CO
    NCLUDERE"
300 GET A$:IF A$="" THEN 300
305 IF A$=CHR$(133) THEN  RUN
310 IF A$=CHR$(136) THEN SYS647
    60
315 GOTO 300
320 REM -----------------------
    ---------
325 REM  ***  ROUTINE ERRORE DI
    SCO  ***
330 REM -----------------------
    ----------
335 INPUT#15,A,A$
340 IF A<>0 THEN PRINTCHR$(147)
    SPC(252)A$:GOTO 280
345 RETURN
350 REM -----------------------
    ---------
355 REM  ** ROUTINE LM DI ANTIR
    ESET **
360 REM -----------------------
    ----------
365 DATA  169,255,133,51,133,55
370 DATA  169,127,133,52,133,56
375 DATA  169,246,141,0,128,141
380 DATA  2,128,169,2,141,1
385 DATA  128,141,3,128,162,5
390 DATA  189,15,253,157,3,128
395 DATA  202,208,247
400 REM -----------------------
    ---------
405 REM  *** ROUTINE LM DI AUTO
    RUN ***
410 REM -----------------------
    ----------
415 DATA  32,94,166
420 DATA  32,142,166,76,174,167
    ,0
425 REM  ----------------------
    ---------
430 REM   ******   VETTORI   *
    ******
435 REM  ----------------------
    ----------
440 DATA  139,227:REM  ** MESS.
    ERRORE **
445 DATA  124,165
450 DATA  26,167:REM  ********
    LIST ****
455 DATA  228,167,134
460 DATA  174,230,40,25,49,76
465 DATA  72,178,0,49,234
470 DATA  246,2:REM  *********
    BREAK (DEFAULT 102,254) ***
    *
475 DATA  71,254:REM  ** INTERR
    .NON MASCH.
480 DATA  74,243,145
485 DATA  242,14,242,80,242,51
490 DATA  243,87,241,202,241
495 DATA  234,246:REM  *******
     STOP (DEFAULT 237,246) ***
    *
500 DATA  62,241,47,243,102,254
505 DATA  46,223:REM  ********
    LOAD *****
510 DATA  237,245:REM  *******
    SAVE *****
```

COME TI CODIFICO IL MESSAGGIO RISERVATO

Gli autori dei romanzi di spionaggio basano spesso le loro trame su oscuri messaggi in codice; anche noi, nel nostro piccolo...

di **Alessandro de Simone**

I primi passi

Esaminando il primo listato ("Primi passi") ci accorgiamo che è facilissimo sviluppare un algoritmo in grado di generare un codice segreto.

La tabella, infatti, si riferisce al primo vettore (A) ed al secondo (B) che contengono l'alfabeto; mentre, però, nel primo vettore l'alfabeto è elencato nel verso giusto, nel secondo è memorizzato nel verso opposto.

L'attento lettore ha già intuito che se associamo ad ogni carattere del primo vettore, il corrispondente carattere del secondo, si perviene alla stesura di un semplice quanto efficace sistema.

Le tre frasi, indicate in fondo alla tabella, vengono visualizzate sul vostro Commodore adoperando il secondo programma ("Seconda fase") che esamina, carattere per carattere, la stringa X$ (riga 180), ne elimina gli spazi (che potrebbero aiutare il "curioso" nella decodifica) e li converte nei corrispondenti codici del vettore B.

Introducendo, con lo stesso programma, la frase codificata, si otterrà, in output, la frase decodificata e perfettamente leggibile, pur se con maggiore difficoltà, a causa dell'assenza di spazi tra una parola e l'altra.

Comodità indispensabile

Uno degli svantaggi di questo (come di qualsiasi altro) sistema, a parte la banalità, è che risulta piuttosto scomodo digitare l'insieme di caratteri codificati per decifrarli.

Naturalmente noi non ci sogneremo mai di inviare un messaggio codificato su carta ma, approfittando del computer, lo invieremo sotto forma di file Ascii (programma n.3: "Terza fase") al nostro misterioso corrispondente. Questi non dovrà fare altro che caricare il programma n.4 ("Quarta fase") ed elaborare il nastro (o il disco) ricevuto.

Se proprio temiamo che il supporto magnetico cada in mani nemiche, niente paura: riempitelo di giochini vari ed innocenti in modo che il file sequenziale passi inosser-

Un messaggio, di solito, è costituito da un

Lo stesso alfabeto, disposto in modo bizzarro, costituisce un codice efficace

Dalla cronaca alla tastiera

Di recente tutti i mass-media si sono occupati di alcuni personaggi che, per sfuggire al controllo della Guardia di Finanza, annotavano introiti illeciti, o in nero, per mezzo di un elaboratore in grado di codificare cifre, nomi ed appunti in modo da offrire una barriera ad occhi indiscreti.

Sembra, tuttavia, che i floppy disk su cui erano annotati gli illeciti valutari siano stati individuati, decodificati e allegati come prove credibili al processo in corso.

In un'epoca come la nostra, popolata da delinquenti comuni, ladri informatici e untori di virus (sempre informatici), la notizia fa certamente scalpore.

Naturalmente ci riferiamo alla sola "tecnica" adoperata per codificare; questa, infatti, considerati gli eventi, sembrerebbe non offrire la "sicurezza" che viene giustamente (!] richiesta non solo per scopi illeciti ma, soprattutto, per impedire ad altri la divulgazione di informazioni riservate (schedari della polizia, dati bancari, segreti militari ed altre amenità dello stesso genere).

L'ingenuità dei romanzieri ci ha fatto sempre pensare che i microfilm di una "formula" segretissima potesse venir racchiuso in un dente finto o in un tacco vuoto.

Sappiamo benissimo, invece, che alcuni progetti richiedono decine di cartelle, di fotocopie, di calcoli che nessuna dentiera, nessuno stivalone riuscirebbe a contenere.

I dati "devono", inoltre, viaggiare rapidamente, magari via modem.

Che fare, allora, per impedire che un prezioso messaggio venga intercettato e, peggio, decodificato?

Noi siamo convinti che, se l'interessato non lo vuole, un messaggio non possa essere decodificato; ed in questo articolo vi spieghiamo perchè.

insieme di caratteri alfabetici (da "a" a "z"), numerici (da 0 a 9), da segni di punteggiatura, accenti, apostrofi e spazi (;:.,?"!').

Se un messaggio viene scritto adoperando tutti i segni indicati e, magari, seguendo un'impaginazione precisa ed agevole, la lettura risulta facilitata e scorrevole.

La leggibilità diminuisce se la punteggiatura lascia a desiderare e se non si adoperano i caratteri maiuscoli.

Nonostante queste limitazioni, tuttavia, un messaggio contiene ancora tutte le informazioni necessarie per una sua completa leggibilità.

Se, però, incominciamo a sostituire nomi propri di persona con soprannomi (il famoso "Antelope Cobbler" dello scandalo Hercules) il reale significato del messaggio diventa chiaro solo a coloro che conoscono la corrispondenza tra soprannome e personaggio.

Continuando con le sostituzioni, si può tentare di utilizzare termini apparentemente innocui, ma dal significato ben preciso e, spesso, ridicolo: "la gatta è saltata sulla mansarda e canta 'o sole mio'".

In casi come questi, pur se non si riesce a decifrare la frase nascosta, si giunge alla certezza che tale frase nasconda necessariamente un secondo significato e, di conseguenza, si tenta di svelarne il mistero o, quantomeno, di seguire con maggiore attenzione le mosse del destinatario del messaggio stesso.

Molto spesso, tuttavia, il significato di alcune frasi è talmente ben celato che nessuno può ragionevolmente insospettirsi perchè, se ciò dovesse avvenire, un qualsiasi messaggio dovrebbe esser passato al setaccio, perfino la pubblicità inviata per posta!

Nel caso di supporti magnetici per computer, poi, la possibilità che un messaggio in codice passi inosservato è elevatissima e, praticamente, ne diventa impossibile la decifrazione.

vato, oppure sia confuso con programma che non si riesce a caricare.

Chi ha fantasia potrà registrare il file Ascii tra un brano musicale e l'altro (caso del registratore a cassette) oppure cancellare il file Ascii dopo averlo registrato (caso del drive): a "recuperarlo" provvederà il destinatario che, avvertito opportunamente, userà uno dei tanti Tool che consentono di recuperare file cancellati accidentalmente.

0	65	A	90	Z	13	78	N	77	M
1	66	B	89	Y	14	79	O	76	L
2	67	C	88	X	15	80	P	75	K
3	68	D	87	W	16	81	Q	74	J
4	69	E	86	V	17	82	R	73	I
5	70	F	85	U	18	83	S	72	H
6	71	G	84	T	19	84	T	71	G
7	72	H	83	S	20	85	U	70	F
8	73	I	82	R	21	86	V	69	E
9	74	J	81	Q	22	87	W	68	D
10	75	K	80	P	23	88	X	67	C
11	76	L	79	O	24	89	Y	66	B
12	77	M	78	N	25	90	Z	65	A

Il caso si complica

Nonostante l'obiettiva difficoltà di decifrazione, non possiamo fare a meno di ammettere che un abile solutore di puzzle può arrivare a determinare il "trucco" dei vettori invertiti.

Si potrebbe complicarlo ulteriormente invertendo l'alfabeto in due gruppi (da "a" a "m" e da "n" a "z") ma chi adopera un computer potrebbe sviluppare, pur se con grande fatica, un numero rilevante di programmi in grado di decodificare il file Asci che si sopetta appartenere alla classe di un "modulo" che, per quanto complesso sia, si fonda pur sempre sulla sequenzialità dell'intero alfabeto.

Se, però, non ci si riferisce all'intero alfabeto, il caso si complica talmente che risulta, a nostro parere, del tutto improbabile la decodifica di un messaggio in codice; a patto di voler evitare di diventare pazzi.

Il programma "Proposta" non è altro che un'idea, da sviluppare adeguatamente, per codificare un qualsiasi messaggio.

Si definisce dapprima il codice (riga 140) con la variabile X$ che potrà essere lunga e varia come si vuole. Subito dopo verrà richiesta (riga 190) la frase da codificare che verrà associata alla stringa Y$.

In segito il programma estrarrà, uno per uno, i caratteri da Y$ (riga 220), scarterà tutti quelli non alfabetici (riga 230) ed effettuerà la differenza tra il codice Ascii Z1 con il codice Ascii Z2 dei caratteri costituenti il codice, scanditi anch'essi uno alla volta.

Se la differenza è inferiore a 65 (codice di "A"), lo stesso valore verrà incrementato di 128, in modo da trasformarlo in maiuscolo.

Dopo aver digitato il programma, fatelo partire e rispondete ai vari Input come nella tabella pubblicata; potete facilmente accorgervi che la frase "Leggete Commodore Computer Club" cambia totalmente aspetto a seconda del codice adoperato ("Commodore", "Alabama", "Ruzzolare", "Aeiouabcd"). Si noti, tuttavia, che adoperando, come codice, un insieme di caratteri appartenenti alla fine dell'alfabeto (come, appunto: "Ruzzolare") la comprensione del messaggio potrebbe essere facilitata. Codici composti dalle prime lettere dell'alfabeto, invece, non lasciano molte possibilità di interpretazione ("Aeiouabcd").

```
IL CODICE DI:    COMMODORE COMPUTER CLUB
CODIFICATO E':   XLNNLWLIUXLNKFGUIXOFY

IL CODICE DI:    TI INVIO UN MILIONE DI LIRE
CODIFICATO E':   GRRMERLFMNRORLMUWRORIU

IL CODICE DI:    TI RACCOMANDO MIO NIPOTE
CODIFICATO E':   GRIZXXLNZMWLNRLMRKLGU
```

Conclusioni

Il sesto ed ultimo listato, come intuitivo, decodifica i messaggi codificati con il programma 5, limitandosi a seguirne il percorso inverso.

Facile è immaginare la miriade di codici, e di codifiche, che è possibile inventare ricorrendo a programmi semplicissimi come quelli pubblicati in queste pagine.

Altrettanto facile è aggiungere righe che possano riversare i messaggi su nastro o disco oppure inviarli via modem ovunque.

Nascondere i file, come già detto, è facilissimo: un vecchio nastro apparentemente dimenticato sul cruscotto della nostra macchina, un disco che sembra aver problemi di lettura, un fastidioso ronzio durante una conversazione telefonica; la fantasia non ha limiti: se lo vogliamo, nessuno riuscirà mai a decodificare i nostri intimi segreti.

Nemmeno i novelli Sherlock Holmes.

L'informatica consente la creazione di messaggi segreti con la massima semplicità

```
con il codice:commodore
il messaggio :leggete commodore computer club
diventa      :jEGGEqECkkMcDaoECkkbihEoCLqB

con il codice:alabama
il messaggio :leggete commodore computer club
diventa      :lEgfehecdmloDorEcnmdutErblib

con il codice:ruzzolare
il messaggio :leggete commodore computer club
diventa      :LEGGEieCkMMODageCkMPUTEgcLqB

con il codice:aeiouabcd
il messaggio :leggete commodore computer club
diventa      :laGGEtdalmigDOrdalmlmfErbjrb

con il codice   : aeiouabcd
il mess.segreto: laGGEtdalmigDOrdalmlmfErbjrb
decodificato e': leggetecommodorecomputerclub
```

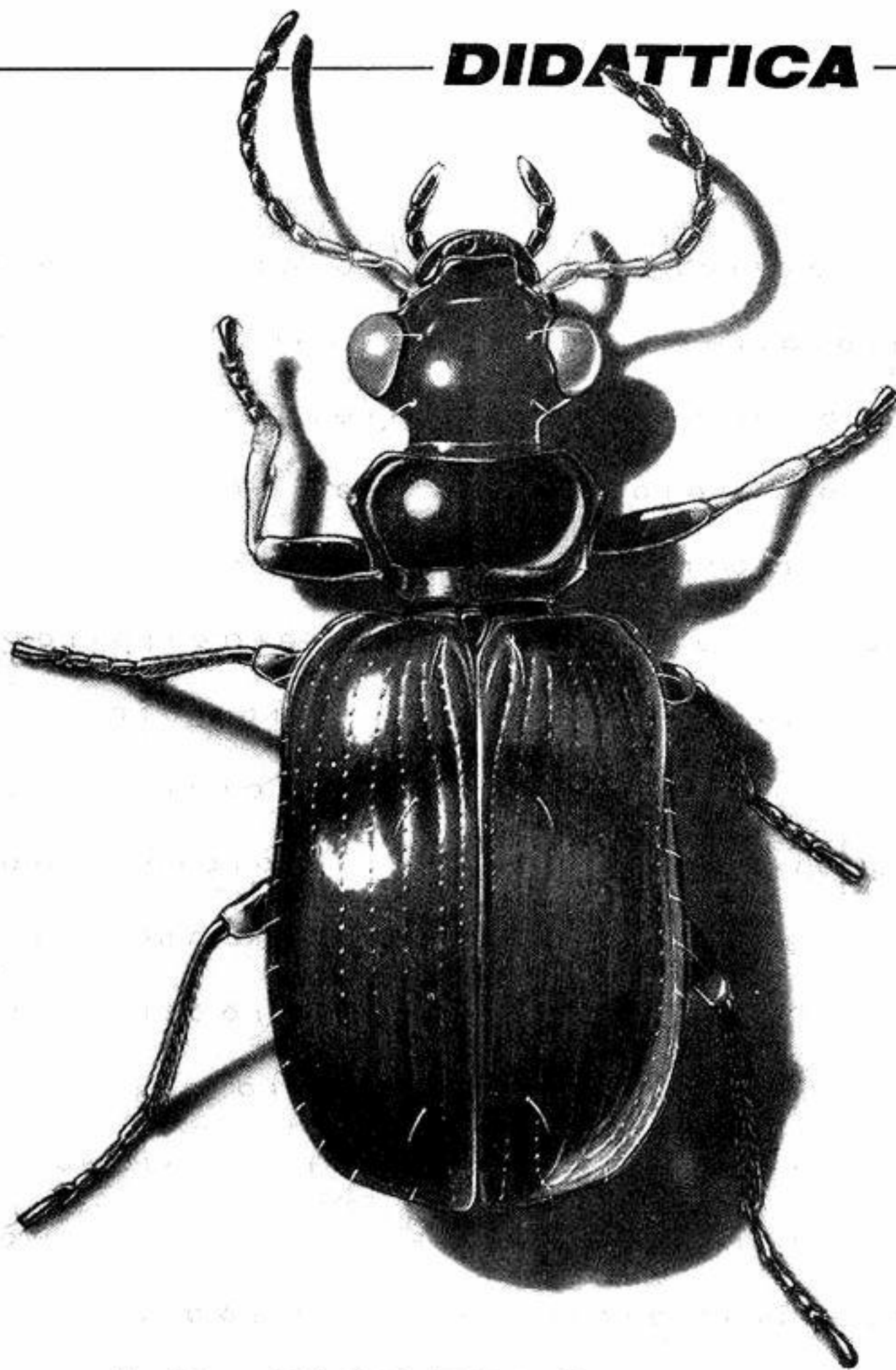
```
100 REM PRIMI PASSI PER LA
110 REM REALIZZAZIONE DI UN
120 REM CODICE SEGRETO:
130 REM I VETTORI INVERTITI
140 :
150 DIM A(25): DIM B(25)
160 FOR I=65 TO 90
170 A(I-65)=I:B(I-65)=90-X
180 X=X+1:NEXT
190 FOR I=0 TO 25
200 PRINT I;A(I);CHR$(A(I));
210 PRINT TAB(15) B(I) CHR$(B(I))
220 NEXT
```

```
100 REM SECONDA FASE:
110 REM CODIFICA E DECODIFICA
120 REM DI UN MESSAGGIO SEGRETO
130 :
140 DIM A(25): DIM B(25)
150 FOR I=65 TO 90
160 A(I-65)=I:B(I-65)=90-X
170 X=X+1:NEXT
180 X$="":INPUT "FRASE";X$: L=LEN(X$)
190 IF L=0 THEN END
200 W$="": FOR I=1 TO L
210 Y$=MID$(X$,I,1): Y=ASC(Y$)
220 IF Y>90 OR Y<65 THEN 240
230 W$=W$+CHR$(B(Y-65))
240 NEXT
250 PRINT"IL CODICE DI:"TAB(15)X$
260 PRINT"E': "TAB(15) W$
270 PRINT:GOTO180
```

```
100 REM TERZA FASE: CODIFICA
110 REM DI UN MESSAGGIO SEGRETO DA
120 REM INVIARE SU SUPPORTO MAGNETICO
130 :
140 DIM A(25): DIM B(25)
150 FOR I=65 TO 90
160 A(I-65)=I:B(I-65)=90-X
170 X=X+1:NEXT
180 INPUT"NASTRO O DISCO (N/D";Q$
190 IF Q$<>"N" AND Q$<>"D" THEN END
200 NP=8: IF Q$="N" THEN NP=1
210 INPUT"NOME FILE";NF$
220 IF NF$="" THEN END
230 IF Q$="D" THEN NF$=NF$+",S,W"
240 OPEN 1,NP,8,NF$
250 X$="":INPUT "FRASE";X$: L=LEN(X$)
260 IF L=0 THEN CLOSE 1: END
270 W$="": FOR I=1 TO L
280 Y$=MID$(X$,I,1): Y=ASC(Y$)
290 IF Y>90 OR Y<65 THEN 310
300 W$=W$+CHR$(B(Y-65))
310 NEXT
320 PRINT"IL CODICE DI:"TAB(15)X$
330 PRINT"E': "TAB(15) W$
340 PRINT#1,W$:PRINT :GOTO 250
```

```
100 REM QUARTA FASE: DECODIFICA
110 REM DI UN MESSAGGIO SEGRETO
120 REM RICEVUTO SU SUPPORTO MAGNETICO
130 :
140 DIM A(25): DIM B(25)
150 FOR I=65 TO 90
160 A(I-65)=I:B(I-65)=90-X
170 X=X+1:NEXT
180 INPUT"NASTRO O DISCO (N/D)";Q$
190 IF Q$<>"N" AND Q$<>"D" THEN END
200 INPUT"NOME FILE";NF$
210 IF NF$="" THEN END
220 IF Q$="N" THEN 250
230 NF$=NF$+",S,R"
240 OPEN 1,8,8,NF$:GOTO260
250 OPEN 1,1,0,NF$
260 GET#1,Y$:IF ST<>0 THEN CLOSE1:END
270 IF Y$="" THEN 260
280 Y=ASC(Y$):IF Y=13 THEN260
290 PRINT CHR$(B(Y-65));:GOTO260
```

```
100 REM PROPOSTA DI SVILUPPO PER
110 REM CODIFICA DI UN
120 REM MESSAGGIO SEGRETO
130 :
140 INPUT"FRASE-CODICE";X$
150 H=LEN(X$):IF H=0 THEN END
160 DIM A(H): FOR I=1 TO H
170 Y$=MID$(X$,I,1):A(I)=ASC(Y$)-65
180 NEXT
190 INPUT"FRASE DA CODIFICARE";Y$
200 L=LEN(Y$):IF L=0 THEN END
210 W$="": U=0: FOR I=1 TO L
220 Z1$=MID$(Y$,I,1): Z1=ASC(Z1$)
230 IF Z1<65 OR Z1>90 THEN 280
240 U=U+1: IF U>H THEN U=1
250 Z2=A(U):Z3=Z1-Z2
260 IF Z3<65 THEN Z3=Z1+128
270 W$=W$+CHR$(Z3)
280 NEXT:PRINTY$
290 PRINT"IL CODIFICE E':":PRINT W$
```

```
100 REM DECODIFICA DI UN
110 REM MESSAGGIO SEGRETO
120 :
130 INPUT"FRASE CODICE";X$
140 H=LEN(X$):IF H=0 THEN END
150 DIM A(H)
160 FOR I=1 TO H
170 Z$=MID$(X$,I,1):A(I)=ASC(Z$)-65
180 NEXT
190 INPUT"FRASE SEGRETA";Y$
200 L=LEN(Y$):IF L=0 THEN END
210 W$="": U=0: FOR I=1 TO L
220 Z1$=MID$(Y$,I,1): Z1=ASC(Z1$)
230 U=U+1: IF U>H THEN U=1
240 Z2=A(U): Z3=Z1+Z2
250 IF Z3>128 THEN Z3=Z1-128
260 W$=W$+CHR$(Z3)
270 NEXT:PRINT
280 PRINT"MESSAGGIO:":PRINT W$
```

LO STRANO CASO DI DOCTOR 64 E MISTER BUG

Oltre al bug dei piccoli Commodore (segnalati sul numero scorso nella rubrica "Domande Risposte"), ve ne sono altri; ed altri ancora...

di **Giancarlo Mariani**

IL CASO SPRITES

Una delle fasi più pesanti che è possibile individuare durante la stesura di un programma, è proprio la digitazione dello stesso.

Generalmente in questa fase si compiono numerosi errori di battitura che, a lavoro completato, portano inevitabilmente ad un non corretto funzionamento dell'elaborato.

In questi casi bisogna ripercorrere l'intero listato e, basandosi anche sui risultati ottenuti, ricercare le linee sospette e correggerle per rimettere le cose a posto.

Fin qui tutto è semplice; ma che fare se il programma, correttamente digitato, non funziona a dovere ma non vi sono errori? Molti di voi penseranno che è impossibile che un programma, senza errori, non funzioni; impossi-

Non sempre un errore è dovuto all'imperizia del programmatore

Non avvilitevi: anche i progettisti sbagliano!

bile, ma vero.

Chi di voi usa correntemente, o ha provato ad usare, il simulatore Gw-Basic per C/64 (prodotto dalla Systems Editoriale) avrà sicuramente notato che, durante una qualsiasi istruzione che coinvolga operazioni su disco (come ad esempio directory, load, save, eccetera), i messaggi corrispondenti ai tasti funzione in basso allo schermo, eventualmente presenti, scompaiono per poi riapparire ad operazione ultimata.

Tale accorgimento non è stato seguito nè per questioni estetiche nè tanto meno pratiche; ma solo perchè, senza "spegnere" le scritte, i suddetti comandi non possono funzionare!

Prima di proseguire è bene sottolineare che, per semplificare la programmazione di Gw-Basic, sono stati usati gli otto sprites del C/64, allineati e posizionati in fondo allo schermo.

Per meglio comprendere quanto verrà detto tra breve, provate ad accendere il C/64 (privo di cartucce come fastload, speeddos e simili) e digitare il seguente programma:

LE PULCI DEL COMPUTER

Quando un programma non sortisce gli effetti desiderati, di solito la colpa è del programmatore.

La storia dell'informatica è piena, tuttavia, di notizie relative a malfunzionamenti strutturali o, in generale, di "base".

Perfino i nomi più illustri non sono sfuggiti ad errori, più o meno noti. Alcune fabbriche di computer, addirittura, sono state costrette a ritirare dal commercio numerosi elaboratori che presentavano difetti incredibili e che hanno spesso contribuito a distruggere la stessa immagine dell'azienda.

I bug (letteralmente: le pulci) presenti in un computer possono essere di duplice natura: software ed hardware. Per gli errori di quest'ultimo tipo non c'è quasi nulla da fare; solo in alcuni casi sporadici è possibile sostituire una parte della componentistica con altra, strutturata in modo diverso. Nella quasi totalità dei casi, invece, l'unica alternativa è disfarsi del computer.

Per i bug software, invece, è sufficiente sostituire il programma (o il sistema operativo) errato con quello corretto, e riprendere il lavoro.

Naturalmente ci riferiamo a programmi e sistemi difettosi per natura e non a copie mal effettuate.

Una ammissione implicita di esistenza di bug, in effetti, è la stessa proposta di versioni successive di programmi, package e interi sistemi operativi.

Con il termine bug, a nostro parere, dovrebbe anche essere indicata la mancanza di comandi, presenti in versioni successive, la cui comodità poteva benissimo esser prevista nella prima versione del package considerato. Se, per esempio, la versione 2 di un certo sistema operativo consente di effettuare copie di file ad una velocità maggiore di quanto consentisse la versione 1, è molto probabile che, in sede di prima stesura, non sia stata prestata la massima attenzione per ottenere subito l'ottimizzazione della velocità.

E' ben vero, tuttavia, che a volte l'idea di un miglioramento viene in mente solo in seguito all'uso pratico ed intensivo dello stesso package.

Ciò che vogliamo affermare, insomma, è che la frenesia di produrre tutto e subito, in un ambiente commerciale fortemente aggressivo e che non lascia respiro, costringe ad alcune sviste, leggere in quasi tutti i casi, pesanti in pochissimi altri.

La differenza tra un bug ed una miglioria, pertanto, si assottiglia e induce ad alcune riflessioni, tra cui:

• *Le versioni Ms-Dos potevano essere in numero minore di quante ne sono oggi disponibili?*

• *La grafica per i Pc IBM e compatibili poteva essere gestita con criterio più razionale fin dall'inizio, oppure è realmente indispensabile confondere gli utenti con la varie schede Hercules, CGA, MDA, EGA e via discorrendo?*

• *L'Amiga 1000 "doveva" nascere così come lo conosciamo? Insomma, il modello 2000 è realmente una naturale evoluzione, oppure rappresenta l'implicita ammissione di qualche svista?*

Ma, come è intuibile, ognuno potrà dare la risposta che crede più opportuna...

```
10 print chr$(147)
20 for k=832 to 896: poke k,255: next
30 for k=2040 to 2047: poke k,13: next
40 for k=53287 to 53294: poke k,1: next
50 a=30: for k=53248 to 53262 step 2: poke k,a: a=a+30: next
60 for k=53249 to 53263 step 2: poke k,100: next
70 poke 53269,255
80 end
```

Il mini programma visualizza otto sprite "pieni" e li posiziona al centro dello schermo, ALLINEANDOLI orizzontalmente.

Prima di farlo partire, registratelo su disco o cassetta per evitare di digitarlo successivamente; ora date il RUN e, con gli otto sprite visualizzati sullo schermo, provate ad eseguire una lettura della directory del disco (LOAD"$",8).

Che cosa succede? Il drive si avvia ed apparentemente la lettura inizia, azione, questa, confermata anche dall'accensione della spia rossa del disco; tale condizione, tuttavia, persisterà sino alla pressione dei tasti Run/Stop e Restore ed allo spegnimento e riaccensione del drive.

Quest'ultimo, infatti, non eseguirà assolutamente ciò che gli è stato ordinato, restando invece inchiodato sino a che un evento esterno (operatore) non modifichi la situazione.

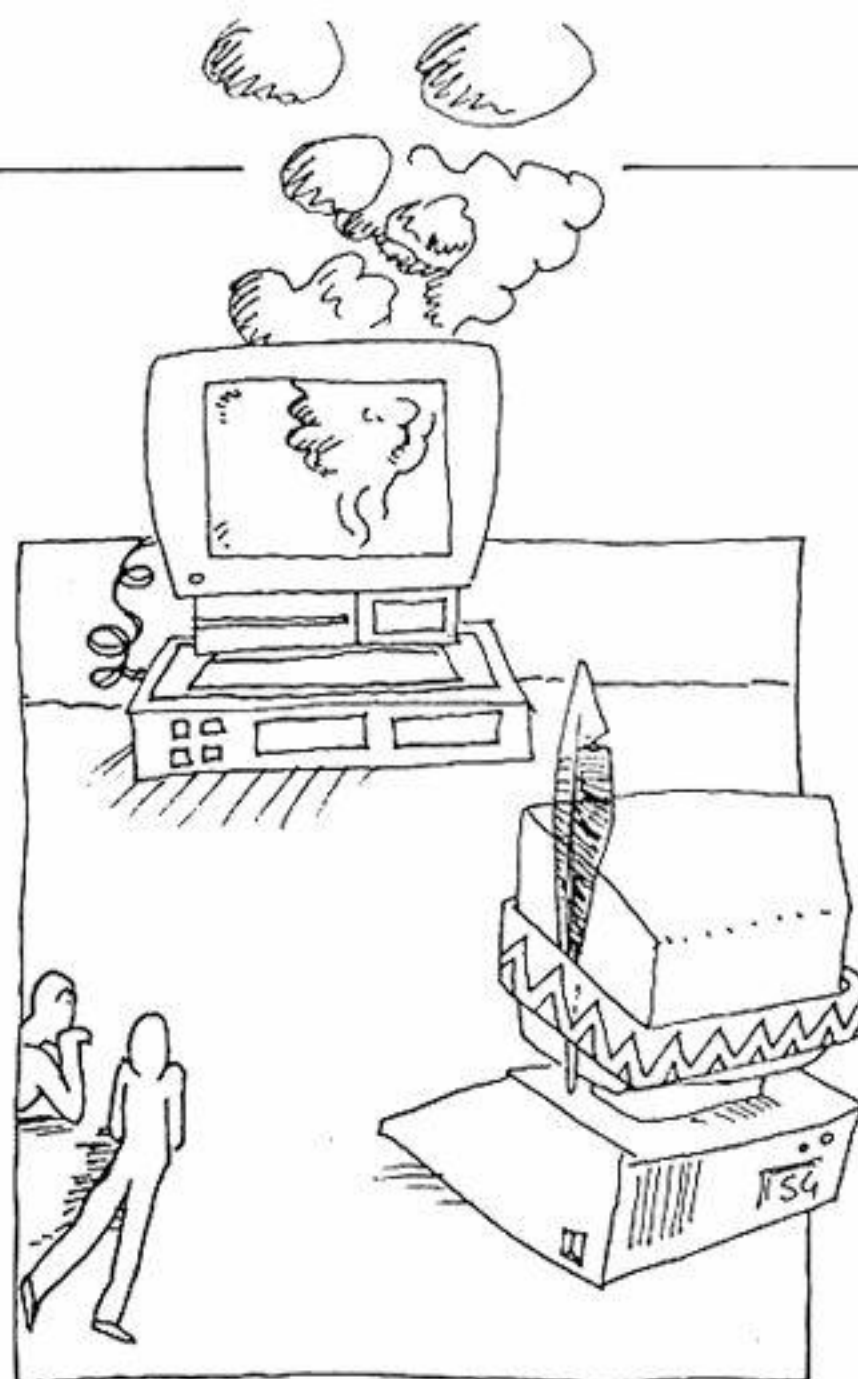
In poche parole: quando sullo schermo del C/64 sono visualizzati DUE (o più) sprites allineati orizzontalmente, qualsiasi tentativo di operazione su disco (a parte i comandi interni del drive) porterà ad un blocco della macchina.

La ragione del fenomeno è probabile che risieda nella gestione dei tempi delle operazioni; probabilmente la visualizzazione degli sprites porta ad un certo rallentamento nelle operazioni di I/O, e di conseguenza il computer non riesce più a sincronizzarsi con il drive nelle fasi di ingresso e di uscita di dati.

L'evento non si verifica se si utilizzano cartucce velocizzatrici del disco, dal momento che queste ultime, generalmente, posseggono al loro interno routines di I/O, diverse da quelle originali, che, forse involontariamente, eliminano il problema.

E' bene ripetere che, posizionando gli sprites a caso sullo schermo, tutto funziona correttamente; il malfunzionamento si verifica solo se gli sprites sono allineati orizzontalmente (provate a cambiarne le coordinate verticali). Speriamo che qualcuno, prima o poi, chiarisca il mistero...

L'AVVENTURA CONTINUA...

Se l'error è in una Rom, purtroppo, non c'è nulla da fare

Passiamo ora ad un altro caso, questa volta risolto, ancora riguardante le operazioni su disco.

Sicuramente saprete il significato dell'istruzione presente nel CBM Basic "STR$". Questa converte un numero in una stringa, consentendone, quindi, la manipolazione, quale incolonnamento, troncamento, arrotondamento, e così via.

L'operazione inversa di STR$ è "VAL", che trasforma una stringa numerica in un numero, sul quale sarà quindi possibile eseguire tutte le operazioni aritmetiche e logiche desiderate. Da quanto detto si deduce che, partendo da un numero, elaborando dapprima STR$ e, subito dopo, VAL, si riottiene lo stesso numero di partenza; ossia...

X

...equivale a...

VAL(STR$(X))

Vediamo ora perchè abbiamo parlato delle due istruzioni Str$ e Val. I comandi su disco si impartiscono assegnando, come numero di periferica, il numero 8; ad esempio:

LOAD "pippo",8

...che DOVREBBE essere uguale ad una istruzione del tipo...

LOAD "pippo",VAL(STR$(8))

...e invece no!

Provando, infatti, ad assegnare, in modo diretto, la seconda istruzione, noterete che il nome del programma da ricercare su disco si trasformerà misteriosamente in "pip 8" (vedi

Consolatevi: anche i computer cosiddetti "professionali" hanno le loro pulci

la scritta "searching for.." che appare sul video) ed ovviamente non sarà più possibile trovarlo sul dischetto.

Effettuate alcune prove, del tipo:

LOAD "nome",val("8ciao ciao")
LOAD "abcdefghi",val("8ciao ciao")

...ed altre istruzioni simili. Noterete che in nessun caso il nome del programma sarà conservato così come l'avete digitato, ma si trasformerà a seconda della stringa inserita nell'istruzione VAL.

La soluzione che chiarisce il mistero, questa volta, è stata trovata: il C/64, quando esegue operazioni con stringhe, pone queste ultime in memoria RAM, comunicando al S.O. l'avvenuta occupazione di determinate celle di memoria in modo da evitare che, con successive elaborazioni di stringhe, riscriva sulla memoria utilizzata in precedenza, distruggendo i dati conservati.

Questo però non succede in caso di operazioni tipo Load o Save. Il S.O., infatti, "prende" il nome del programma, lo deposita in una determinata zona di memoria, ma "dimentica" di proteggerla da sovrascritture.

Infatti, durante la successiva operazione su stringhe (nel nostro caso: VAL) il C/64 riterrà libera l'area in cui ha, invece, trascritto il nome del programma, e la userà per scrivere la stringa contenuta in VAL, alterando, pertanto, il nome del programma.

I puntatori al nome, invece, resteranno inalterati, con il conseguente malfunzionamento che si verificherà quando il S.O. cercherà i dati scritti in precedenza.

Questo è uno dei tanti "bug" presenti in qualsiasi sistema operativo: naturalmente poteva essere corretto aggiungendo un paio di istruzioni LM, ma evidentemente i progettisti del C/64 non se ne sono accorti, o se ne sono accorti troppo tardi per porvi rimedio.

GRAN FINALE

Ed ecco un terzo "bug", questa volta che riguarda operazioni non su disco ma sulla memoria. Come certamente saprete, la memoria del C/64 è composta da circa 20k di ROM e 64k di RAM, di cui un certo numero (esattamente 16) è stata posta "sotto" le Rom del S.O. e dell'interprete Basic per motivi tecnici.

Per leggere i dati contenuti nella RAM "nascosta" bisogna, solo tramite LM, disabilitare il S.O. Per svolgere tale compito (disabilitare le Rom del S.O. e/o dell'interprete Basic) è necessaria la manipolazione di alcuni bit della locazione 1:

bit 0: ROM BASIC (0=off, 1=on)
bit 1: ROM S.O. (0=off, 1=on)

Gli altri bit del byte 1 non interessano al nostro scopo.

Per leggere un solo dato nella Ram "sottostante" la Rom, le operazioni da svolgere dovrebbero quindi essere:

- *Disabilitare la ROM*
- *Leggere il dato*
- *Riabilitare la ROM.*

Eseguendo le operazioni suddette, invece, non si otterrà alcun risultato tranne quello di leggere un dato totalmente privo di significato, ossia non il dato presente nella locazione desiderata della RAM, ma un dato che il computer recupera chissà dove.

Questo perchè i 16k di RAM posti sotto la rom, devono essere "abilitati" prima di leggerli: dopo aver disabilitato la ROM tramite la locazione 1, pertanto, va eseguita una lettura "a vuoto" di un dato in una locazione qualsiasi del banco di 8k di ram abilitato.

Tale lettura servirà solamente a "risvegliare" la RAM, che sarà quindi disponibile per tutte le altre operazioni desiderate.

Una lettura preventiva, insomma, deve essere effettuata ogni volta che si accede alla RAM modificando la locazione 1. Omettendo l'operazione descritta, infatti, il primo dato letto dalla RAM andrà perso (viene, infatti, utilizzato per il "risveglio"), con le conseguenze che si possono immaginare.

Riassumendo, le operazioni da seguire per leggere un dato posto nella RAM nascosta sono:

- *Modificare opportunamente la locazione 1.*
- *Leggere un dato da una locazione qualsiasi del banco da 8k abilitato.*
- *Leggere i dati desiderati*
- *Riabilitare la ROM tramite la locazione 1*

Anche questo mistero è rimasto sinora irrisolto, pur se riteniamo che la prima lettura dalla RAM abiliti un particolare circuito (quale?) che commuta il trasferimento delle informazioni dalla ROM alla RAM, che altrimenti resterebbe disabilitato.

I bugs descritti sono stati individuati durante la stesura del simulatore Gw-Basic; la conoscenza degli stessi può essere certamente utile soprattutto per il programmatore LM; se qualche lettore dovesse scoprire altri bugs, oppure trovasse la soluzione per quelli irrisolti, può comunicarlo alla redazione di Commodore Computer Club, che provvederà poi a diffonderla tramite la rivista.

Buona ricerca...

SCHERMAO MERAVIGLIAO

Che cos'è mai lo schermo di un C/64 che non può visualizzare contemporaneamente l'alta risoluzione e la pagina testo?: è come un video senza visione...

di **Gianmario Pianta**

Prima di addentrarci nei particolari, descriveremo il funzionamento del nuovo comando proposto osservandone la sintassi specifica:

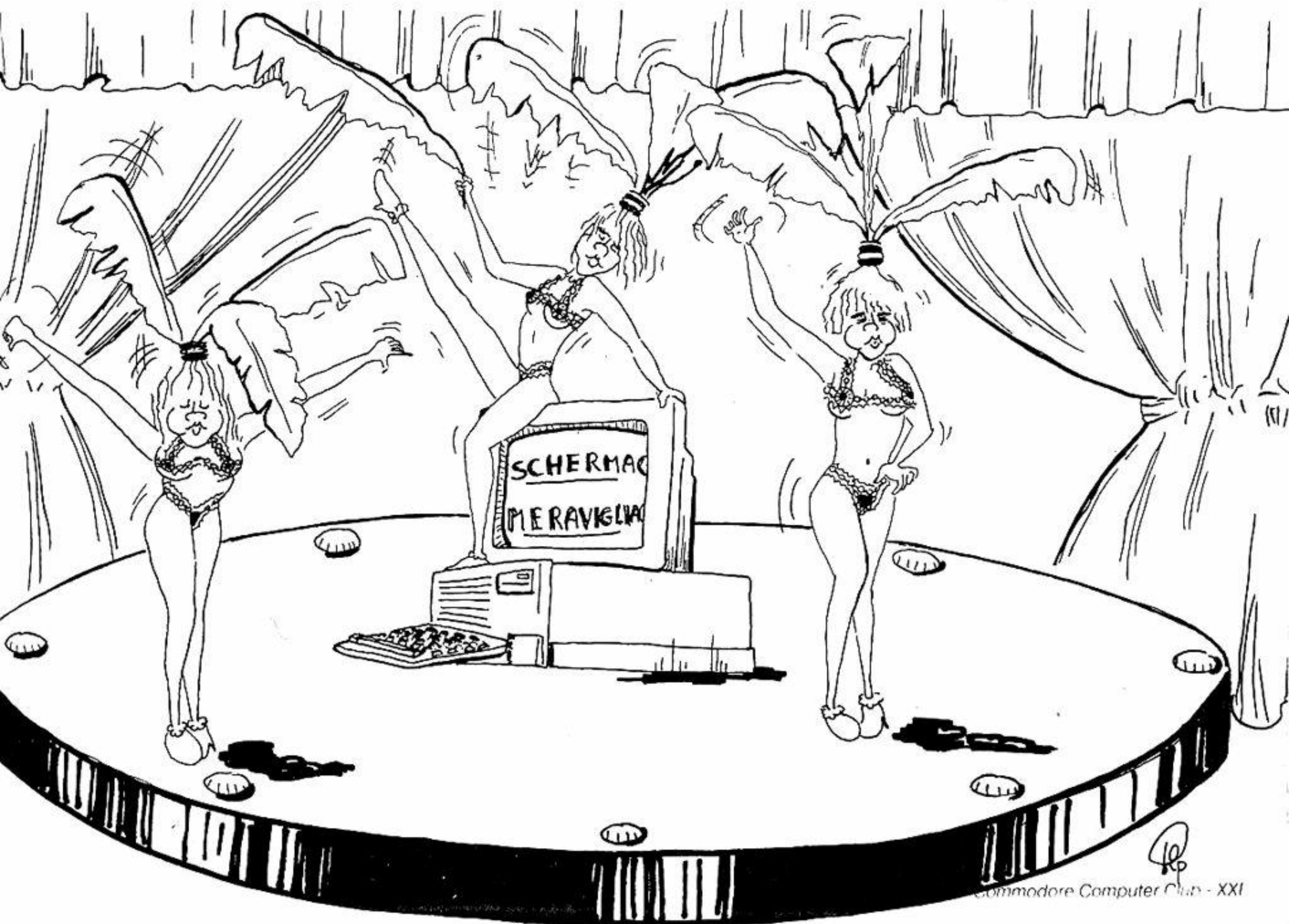
SYS 49345, E, [M, [S, [L, [F]]]]

Le lettere usate hanno un significato ed un range (:campo d'azione) specifico che ora esamineremo in particolare:

E = Enable ON OFF pagina grafica (0-1)
M = Multicolor ON OFF (0-1)
S = Switch UP DOWN testo (0-1)
L = Line interrupt (0-201)
F = Fill graphic memory (0-255)

E: Enable

Serve per attivare (1) ovvero disattivare (0) la pagina grafica. Esempio:
SYS 49345,1

Una pagina hi-res spesso è insufficiente per rappresentare altri dati

Mescolare grafica e testo si rivela utile in molte occasioni

...attiva, mentre...
SYS49345,0
...disattiva

M: Multicolor
Anche questo è un flag per attivare o disattivare il modo multicolore; esempi:
SYS 49345,1,1
...imposta pagina grafica in multicolor, mentre...
SYS 49345,1,0
...imposta la pagina grafica standard.

S: Switch
Analogo ai due precedenti parametri, "switch" ha la funzione di stabilire se scrivere sopra o sotto (da intendere come alto e basso) la pagina grafica:
SYS 49345,1,0,0
...predispone up write, mentre...
SYS 49345,1,0,1
...predispone down write.

L: Line
Questo parametro stabilisce lo spessore (misurato in pixels) dello schermo in modo testo, o in modo hires, a seconda che lo switch sia impostato, rispettivamente, a zero (UP) ovvero a uno (down).
SYS 49345,1,0,0,100
...predispone per il testo la metà alta del video ed in modalità hires la metà sottostante.

F: Fill
Questo parametro, se specificato, ha l'effetto di riempire la pagina grafica con il numero digitato:
SYS 49345,1,0,0,0,0
...produce un clear della pagina.
Se il parametro Fill non è specificato, la pagina grafica rimane tale e quale, cioè rimangono presenti gli eventuali dati caricati in precedenza.

Il nuovo comando funziona sia in modo diretto che da programma ed i parametri inclusi nelle parentesi quadre (come ormai dovreste sapere) possono anche non essere specificati.
Se dovesse capitare che il programma si inchiodi, sarà necessario premere Run/Stop Restore e riattivarlo con SYS 49438, ma per evitare questi spiacevoli inconvenienti è buona norma usare questi tasti ogni qualvolta il programma non serve.

Il programma nei dettagli

Il programma risulta suddiviso in due routine principali e tre soubroutine. Le prime sono indicate con il nome "Raster" e "Command", mentre le altre con "Testo Mode", "Hires Mode" e "Get Flag".

La routine Raster è quella che decide quando è giunto il momento di cambiare la risoluzione dello schermo chiamando una delle due soubroutine dedicate; inoltre controlla anche il flag di Enable, stabilendo così se deve essere attivo o disattivo lo Split.
Le due soubroutine Testo e Hires servono, come si può capire dal nome stesso, per cambiare la risoluzione dello schermo. In più Hires ha anche il controllo del flag di Multicolor.
La routine Command è invece quella dedicata all'interpretazione del nuovo comando ed è quindi incaricata di prelevare i parametri e predisporli per l'uso.
Get flag è il nome dell'ultima soubroutine che viene usata dalla routine Command. La sua funzione è quella di prelevare i parametri che hanno range 0-1 e segnalare eventuali errori.

I listati pubblicati

Il programma vero e proprio in linguaggio macchina può esser caricato servendosi del listato Basic "caricatore" che, sfruttando i comandi Read e Data, alloca correttamente i valori in memoria.
E' inutile ribadire il concetto di registrare il tutto prima di impartire la Sys.
E' presente anche un programma dimostrativo in Basic, da lanciare DOPO aver caricato e lanciato la routine l.m., che servirà a far meglio comprendere l'utilizzo del nuovo comando nelle varie forme sintattiche che possono essere usate.

Sul dischetto "Directory" di questo mese è anche presente un file (Hires data) che rappresenta un disegno in alta risoluzione da caricare prima del dimostrativo (con la sintassi ,8,1) per vedere qualcosa al posto della scarna pagina grafica che si presenta all'accensione del computer.

Logicamente coloro che digiteranno il listato dovranno provvedere a caricare un'immagine in loro possesso oppure a crearne una servendosi di uno dei numerosi tool grafici in circolazione.

Per usare il comando "Split" con altri programmi basic, è sempre necessario, dopo averlo caricato e fatto partire, disabilitarlo prima di caricare il programma che lo userà; questo, pertanto, dovrà avere una linea contenente l'ormai famosa SYS 49438.

Raster-mania

Alla base di qualunque tecnica grafica è presente il Raster Register

Il programma di queste pagine permette di visualizzare, contemporaneamente, una parte di schermo in modo testo e l'altra in alta risoluzione, permettendo anche di passare al multicolor.

E' perfino possibile scegliere direttamente lo "spessore" della schermata in alta e bassa risoluzione e, se non bastasse, si può effettuare anche un fill (riempimento della pagina grafica) con un valore qualsiasi.

Sono infatti disponibili varie opzioni, grazie ad un nuovo comando che riassume altrettante funzioni, mediante una sintassi semplicissima...

SYS addr, parametro, parametro, ...

...di cui si parla diffusamente nell'articolo.

Ma vediamo ora di capire come può funzionare un programma del genere. Non poteva certo mancare la solita routine in l.m. che controlla le interruzioni del Raster Register.

La sua manipolazione, infatti, permette all'operatore di scegliere direttamente la linea di schermo sulla quale provocare l'interruzione che genera la netta divisone tra le due pagine.

Per gestire una sola interruzione (anzichè due) la routine possiede un controllo che verifica se l'operatore ha impostato il testo al di sopra della grafica o viceversa; nel comando, quindi, sarà necessario specificare tale parametro.

Il funzionamento del programma è veramente banale: si tratta solo di fissare un'interruzione video e di controllare l'altra in modo da provocare il cambiamento di schermo. In pratica si tratta di passare al computer la seguente informazione:

Se il raster (ovvero il pennello elettronico che disegna sul video) ha raggiunto la linea indicata dall'operatore, allora setta la pagina grafica e prepara la prossima interruzione; altrimenti setta la pagina testo e prepara l'interruzione sulla linea prefissata.

Semplice(!) vero? Bè, a parte gli scherzi non è che sia proprio difficile, solo che bisogna comunicare con il computer utilizzando il suo linguaggio.

La SYS di attivazione della routine vale 49438, mentre la SYS di "chiamata" per il passaggio dei parametri è 49345.

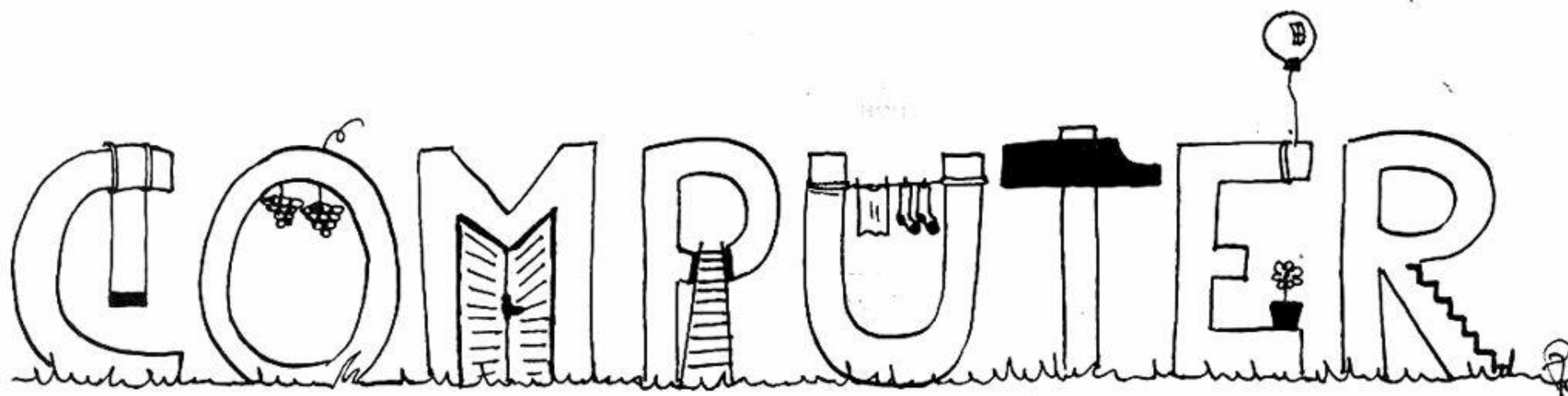
SYS 49438 è da digitare, insomma, ogni qualvolta il programma viene resettato sia tramite un reset hard/soft (Sys 64738 o pulsante) che mediante la pressione contemporanea dei tasti Run/Stop e Restore.

E' molto importante, prima di effettuare un caricamento, disabilitare in uno dei due modi il programma in questione, pena il possibile aborto della routine stessa di caricamento del kernal che provvederà a bloccare il funzionamento del drive fino al successivo reset di sistema (coma sopra descritto).

Questo è l'unico difetto, se così vogliamo chiamarlo, che il programma presenta, tralasciando il necessario rallentamento del basic, introdotto per ottenere una certa stabilità in corrispondenza della lina di interruzione, evitando in tal modo il fastidioso sfarfallio nella divisione dei due (per così dire) schermi.

```
1 REM SPLIT SCREEN PER C/64
2 REM CARICAMENTO CODICI MACCHINA
3 :
10 POKE 53281,0:POKE53280,0
20 PRINTCHR$(147):PRINT"SPLIT C-64"
30 PRINT"ATTENDERE IL NUMERO 398
40 FOR T=49152 TO 49550:READX:Y=Y+X
45 PRINT CHR$(19)UU:UU=UU+1
50 IF X<0 OR X>255 THEN 2000
60 POKE T,X:NEXT
70 IF Y<> 51330 THEN 2020
80 PRINT CHR$(147)"ABILITARE CON:"
90 PRINT "SYS 49438"
100 DATA 169,000,141,026,193,141,027
110 DATA 193,141,028,193,169,032,141
120 DATA 029,193,169,046,141,020,003
130 DATA 169,192,141,021,003,173,017
140 DATA 208,041,127,141,017,208,173
150 DATA 029,193,141,018,208,169,129
160 DATA 141,026,208,096,173,025,208
170 DATA 041,001,240,249,141,025,208
180 DATA 173,026,193,240,023,173,018
190 DATA 208,201,016,176,022,169,016
200 DATA 141,018,208,173,028,193,208
210 DATA 006,032,137,192,076,049,234
220 DATA 032,118,192,076,049,234,205
230 DATA 029,193,176,229,173,029,193
240 DATA 141,018,208,173,028,193,208
250 DATA 006,032,118,192,076,188,254
260 DATA 032,137,192,076,188,254,173
270 DATA 017,208,041,223,141,017,208
280 DATA 169,021,141,024,208,169,200
290 DATA 141,022,208,096,173,027,193
300 DATA 208,008,169,200,141,022,208
310 DATA 076,155,192,169,216,141,022
320 DATA 208,173,017,208,009,032,141
330 DATA 017,208,169,029,141,024,208
340 DATA 096,032,253,174,032,158,183
350 DATA 224,002,016,004,032,121,000
360 DATA 096,169,000,141,026,193,162
370 DATA 014,108,000,003,032,169,192
380 DATA 142,026,193,201,044,240,001
390 DATA 096,032,169,192,142,027,193
400 DATA 201,044,240,001,096,032,169
410 DATA 192,142,028,193,201,044,240
420 DATA 001,096,032,253,174,032,158
430 DATA 183,138,024,105,049,201,251
440 DATA 176,199,141,029,193,032,121
450 DATA 000,201,044,240,001,096,032
460 DATA 253,174,032,158,183,138,162
470 DATA 000,134,250,162,032,134,251
480 DATA 160,000,145,250,136,208,251
490 DATA 230,251,166,251,224,064,208
500 DATA 243,096,000,000,000,000,162
510 DATA 000,142,032,208,142,033,208
520 DATA 162,001,142,134,002,032,068
530 DATA 229,162,039,189,102,193,056
540 DATA 233,165,157,040,004,202,208
550 DATA 244,173,039,193,201,110,240
560 DATA 011,160,255,136,208,253,238
570 DATA 039,193,076,038,193,169,014
580 DATA 141,032,208,141,134,002,169
590 DATA 006,141,033,208,169,001,141
600 DATA 039,193,032,068,229,076,000
610 DATA 192,197,184,181,177,174,185
620 DATA 197,181,183,180,172,183,166
630 DATA 178,197,167,190,197,181,174
640 DATA 166,179,185,166,197,172,174
650 DATA 166,179,178,166,183,174,180
660 DATA 197,184,180,171,185,197,004
670 END
2000 PRINT"IN LINEA:"
2010 PRINT PEEK(63)+256*PEEK(64)
2020 PRINT"C'E' UN ERRORE":END
```

```
$C000 LDA#$00      ---------------------
$C002 STA$C11A     resetta la tabella
$C005 STA$C11B     di memoria per tutti
$C008 STA$C11C     i parametri
$C00B LDA#$20      e prepara l'interrupt
$C00D STA$C11D     ---------------------
$C010 LDA#$2E      fa puntare al vettore
$C012 STA$0314     di interrupt la
$C015 LDA#$C0      routine a C02E.
$C017 STA$0315     ---------------------
$C01A LDA$D011     prepara la prima
$C01D AND#$7F      interruzione del
$C01F STA$D011     raster alla linea
$C022 LDA$C11D     $20 #32
$C025 STA$D012     ---------------------
$C028 LDA#$81      attiva IRQ di RASTER
$C02A STA$D01A     ---------------------
$C02D RTS          ritorna al BASIC
$C02E LDA$D019     ---------------------
$C031 AND#$01      attende una IRQ video
$C033 BEQ$C02E     ---------------------
$C035 STA$D019     resetta flag IRQ
$C038 LDA$C11A     controllo flag ENABL
$C03B BEQ$C054     se = 0 disabilita
$C03D LDA$D012     legge RASTER
$C040 CMP#$10      compará con linea 16
$C042 BCS$C05A     e' > ? se si' salta
$C044 LDA#$10      prepara interruzione
$C046 STA$D012     sulla linea 16
$C049 LDA$C11C     controllo UP/DOWN
$C04C BNE$C054     se = DOWN salta
$C04E JSR$C089     soub. HIRES MODE
$C051 JMP$EA31     scandisce tastiera
$C054 JSR$C076     soub. TEXT MODE
$C057 JMP$EA31     scandisce tastiera
$C05A CMP$C11D     controlla LINE INTER
$C05D BCS$C044     se > allora salta
$C05F LDA$C11D     prepara interruzione
$C062 STA$D012     a LINE INTERRUPT
$C065 LDA$C11C     controllo UP/DOWN
$C068 BNE$C070     se = DOWN salta
$C06A JSR$C076     soub. TEXT MODE
$C06D JMP$FEBC     esce dall'interrupt
$C070 JSR$C089     soub. HIRES MODE
$C073 JMP$FEBC     esce dall'interrupt
```

```
1 REMPROGRAMMA DIMOSTRATIVO
2 REMSPLIT TESTO E GRAFICA HI-RES PER C/64
3 :
4 REME' INDISPENSABILE CARICARE E LANCIARE IL PROGRAMMA LM (VEDI TESTO)
5 :
10 PRINT"":SYS49438
20 FORT=0TO200:SYS49345,1,0,0,T:NEXT
30 FORT=200TO0STEP-1:SYS49345,1,0,0,T:NEXT
40 FORT=200TO0STEP-1:SYS49345,1,0,1,T:NEXT
50 FORT=0TO200:SYS49345,1,0,1,T:NEXT
60 SYS49345,1,0,0,9
70 PRINT" SPLIT:    TESTO E GRAFICA INSIEME.....":FORT=0TO2000:NEXT
80 SYS49345,1,0,0,100
90 PRINT"QUESTO E' UN ESEMPIO :"
100 PRINT"CONTEMPORANEAMENTE TESTO (SOPRA)"
110 PRINT"GRAFICA HI-RES   (SOTTO)"
120 FORT=0TO2000:NEXT:PRINT""
130 SYS49345,1,0,1
140 PRINT"OPPURE INVERTIRE..."
150 PRINT"GRAFICA (SOPRA) E TESTO (SOTTO)
160 FORT=0TO2000:NEXT
180 PRINT"ED ORA.......... TUTTO IN HI-RES":FORT=0TO1500:NEXT:PRINT""
190 SYS49345,1,0,1,200
200 FORT=0TO1500:NEXT
210 SYS49345,1,0,1,192
212 P$=""
214 PRINTP$"PUOI ENTRARE IN MULTICOLOR......";:SYS49345,1,1
216 FORT=0TO1000:NEXT
218 PRINTP$"O TORNARE AL MODO STANDARD HI-RES";:SYS49345,1,0
220 FORT=0TO1000:NEXT:PRINTP$"E TORNARE AL MODO TESTO..........";
230 FORT=0TO1500:NEXT
240 SYS49345,1,0,1,176:PRINT""
250 PRINT"E' TUTTO CHIARO?......";
260 FORT=176TO0STEP-1:SYS49345,1,0,1,T:NEXT:FORT=0TO1000:NEXT:PRINT""
270 SYS49345,1,0,0,68
280 PRINT"PUOI CANCELLARE LO SCHERMO HI-RES...":FORT=0TO750:NEXT
290 SYS49345,1,0,0,68,0
300 PRINT"O RIEMPIRLO CON UN VALORE:"
310 FORT=0TO1000:NEXT:PRINT"NUMERO    : 155":FORT=0TO500:NEXT
320 SYS49345,1,0,0,68,155
330 FORT=0TO1000:NEXT:PRINT"NUMERO    : 128":FORT=0TO500:NEXT
340 SYS49345,1,0,0,68,128
350 FORT=0TO750:NEXT:PRINT"*****   FINE....   *****"
360 FORT=0TO2000:NEXT:SYS49345,0
370 PRINT"";:END
```

COMMAND :

```
$C0C1 JSR$C0A9      soub. GET FLAG
$C0C4 STX$C11A      memorizza ENABLE
$C0C7 CMP#$2C       controllo virgola
$C0C9 BEQ$C0CC      se c'e' salta
$C0CB RTS           ritorna al BASIC
$C0CC JSR$C0A9      soub. GET FLAG
$C0CF STX$C11B      memorizza MULTICOLOR
$C0D2 CMP#$2C       controllo virgola
$C0D4 BEQ$C0D7      se c'e' salta
$C0D6 RTS           ritorna al BASIC
$C0D7 JSR$C0A9      soub. GET FLAG
$C0DA STX$C11C      memorizza SWITCH
$C0DD CMP#$2C       controllo virgola
$C0DF BEQ$C0E2      se c'e' salta
$C0E1 RTS           ritorna al BASIC
$C0E2 JSR$AEFD      salta la virgola
$C0E5 JSR$B79E      preleva LINE INTER.
$C0E8 TXA           parametro in ACC.
$C0E9 CLC           pulisce il CARRY
$C0EA ADC#$31       somma 49 decimale
$C0EC CMP#$FB       risultato > 251 ?
$C0EE BCS$C0B7      se si' salta a errore
$C0F0 STA$C11D      memorizza LINE INTER.
$C0F3 JSR$0079      preleva prossimo CHR$
$C0F6 CMP#$2C       e' una virgola ?
$C0F8 BEQ$C0FB      se si' salta
$C0FA RTS           ritorna al BASIC
$C0FB JSR$AEFD      salta la virgola
$C0FE JSR$B79E      preleva FILL
$C101 TXA           parametro in ACC.
$C102 LDX#$00       ---------------------
$C104 STX$FA        prepara $FA e $FB per
$C106 LDX#$20       puntare all'inizio
$C108 STX$FB        della pagina grafica
$C10A LDY#$00       ---------------------
$C10C STA($FA),Y    cancella un blocco da
$C10E DEY           255 locazioni
$C10F BNE$C10C      ---------------------
$C111 INC$FB        incrementa HI BYTE e
$C113 LDX#$FB       controlla se a
$C115 CPX#$40       finito il FILL
$C117 BNE$C10C      ---------------------
$C119 RTS           ritorna al BASIC
```

TABELLE DI MEMORIA :

```
$C11A - ENABLE
$C11B - MULTICOLOR
$C11C - SWITCH
$C11D - LINE INTERRUPT
```

TEXT MODE :

```
$C076 LDA$D011      ---------------------
$C079 AND#$DF       disabilita HIRES
$C07B STA$D011      ---------------------
$C07E LDA#$15       rimette a posto banco
$C080 STA$D018      ---------------------
$C083 LDA#$C8       disbilita multicolore
$C085 STA$D016      ---------------------
$C088 RTS           ritorna alla routine
```

HIRES MODE :

```
$C089 LDA$C11B      controllo MULTICOLORE
$C08C BNE$C096      se ON allora salta
$C08E LDA#$C8       disabilita multicolor
$C090 STA$D016      ---------------------
$C093 JMP$C09B      salta a settare HIRES
$C096 LDA#$D8       abilita multicolore
$C098 STA$D016      ---------------------
$C09B LDA$D011      abilita la pagina
$C09E ORA#$20       grafica e...
$C0A0 STA$D011      ---------------------
$C0A3 LDA#$1D       sceglie il banco
$C0A5 STA$D018      ---------------------
$C0A8 RTS           ritorna alla routine
```

GET FLAG :

```
$C0A9 JSR$AEFD      controllo virgola
$C0AC JSR$B79E      prelievo parametro
$C0AF CPX#$02       e' => 2 ?
$C0B1 BPL$C0B7      se si' salta a errore
$C0B3 JSR$0079      preleva prossimo CHR$
$C0B6 RTS           ritorna alla routine
$C0B7 LDA#$00       disabilita SPLIT
$C0B9 STA$C11A      ---------------------
$C0BC LDX#$0E       visualizza l'errore
$C0BE JMP($0300)    ILLEGAL QUANTITY?
```

UNO SPRITE CON UNA MARCIA IN PIU'

Il movimento di uno sprite è un'operazione che, con il Basic, non raggiunge velocità elevate; ma in linguaggio macchina...

di **Domenico Pavone**

Dal momento che, nella programmazione in Basic, il comando POKE è quello che più si presta ad essere paragonato alle istruzioni di manipolazione dei registri A (Accumulatore), X ed Y, per operare un valido raffronto tra i due linguaggi e cominciare ad entrare nella logica dell'Assembly, usiamo come punto di partenza il breve programma del listato 1.

Come farlo in Basic

Questo si occupa, in maniera piuttosto semplice per facilitarne il confronto, di far scorrere sullo schermo, in senso orizzontale, uno sprite piuttosto rudimentale, formato da un rettangolo bianco "pieno". Come potete notare, il listato è diviso in due parti dalla REM di linea 90: le righe in alto si occupa-

Un programma l.m. può essere caricato facilmente con un listato Basic

La facilità di programmazione del Basic si paga con una lentezza esecutiva

no della definizione delle caratteristiche dello sprite, mentre quelle da 100 in poi sovrintendono al suo movimento.

Senza stare ad analizzare in dettaglio le procedure necessarie allo scopo (peraltro specificate dalle numerose REM) ricordiamo solo che se nel movimento orizzontale di uno sprite si supera la posizione 255 dello schermo, è necessario intervenire sulla locazione 53264, ponendone a 1 il bit corrispondente allo sprite attivato (nel nostro caso lo sprite 0).

In effetti le varie POKE del programma andrebbero espresse più correttamente adoperando gli operatori logici AND e OR, ma visto che la loro reale funzione è strettamente legata al linguaggio macchina, affronteremo meglio l'argomento quando avremo acquisito una certa consuetudine alla notazione binaria.

Ma torniamo al nostro mini programma.

Dopo averlo digitato e salvato su supporto magnetico, lanciamolo col classico RUN, mentre, per fermarlo, sarà necessario premere Run/Stop (+ Restore, se volete "resettare" anche lo sprite).

Si vedrà l'animazione percorrere lo schermo ad una velocità che, per quanto si ottimizzi il listato (minor numero di righe, esclusione delle REM, e così via), non potrà essere incrementata più di tanto.

Inoltre, se si osserva il lato destro del video, si noterà, ad ogni passaggio, un breve lampeggiamento dell'immagine, causata dal tempo necessario al Basic per eseguire l'istruzione di linea 150. Se invece del Basic usiamo però il linguaggio macchina (l.m.), il discorso cambia, e non di poco.

Per vedere anche le interazioni possibili tra i due linguaggi, proveremo, dunque, a realizzare una routine l.m. che gestisca il movimento dello sprite.

Per fare ciò digitate, registrate e attivate (con il solito Run) il listato n.2.

Quest'ultimo rappresenta, infatti, la sequenza di codici numerici (apparentemente privi di significato), che fornirà al microprocessore le necessarie istruzioni per svolgere il compito che ci prefiggiamo: in altre parole, un codice in linguaggio macchina.

Se non siete ancora in possesso del Macro Assembler Commodore, che è estremamente facile (ed economico) da reperire, l'unica strada possibile è quella di copiare e mandare in esecuzione il listato 2, che provvede, appunto, a caricare la routine nelle locazioni da 49152 in poi, grazie ai comandi Read, Data e Poke.

In tal modo, tuttavia, ci si limita ad un compito "passivo", senza la possibilità di personalizzare, in seguito, la routine che, con le dovute modifiche, potrebbe rivelarsi molto utile se inserita in altri programmi più "evoluti" dell'esempio proposto. Questo salto di qualità è invece consentito dal Macro Assembler, il quale però necessita di un certo "addestramento" all'uso.

In altre parole, il listato 2 consente, lavorando in Basic, di "preparare il terreno" al programma 3 che, grazie al comando Sys, attiverà la routine.

```
1 REM ***                          ***
2 REM ***   L I S T A T O   1      ***
3 REM ***                          ***
4 :
20 PRINTCHR$(147):VC=53248:Y=150
30 FORI=0TO62
40 POKE832+I,255:REM -> DEF. SPRITE
50 NEXT
60 POKE2040,13:REM ---> PUNTATORE
70 POKEVC+1,Y: REM ---> COORD. VERT.
80 POKEVC+39,1:REM ---> COLORE
90 REM---------------------------
100 POKEVC+21,1:REM ---> ATTIVAZIONE
110 POKEVC+16,0:REM ---> POS. X<255
120 FORX=0TO255
130 POKEVC,X:    REM ---> MOVIMENTO
140 NEXT
150 POKEVC+16,1:REM ---> POS. X>255
160 FORX=1TO88
170 POKEVC,X:    REM ---> MOVIMENTO
180 NEXT:GOTO110
```

Più veloci della luce

Ma vediamo finalmente il risultato dei nostri sforzi.

Se avete qualche dubbio, attuate le seguenti fasi:

- Spegnete e riaccendete il computer (in modo da eliminare ogni "residuo" di eventuali errori).

- Caricate e lanciate il programma 2 (in modo da "depositare", grazie al Basic, il programma in l.m. che, lo ripetiamo, è rappre-

sentato dalla successione di numeri memorizzati nelle locazioni di memoria Ram numerate da 49152 a 49208).

• Caricate il programma n.3.

• Se qualcosa non va per il verso giusto vuol dire che non avete trascritto correttamente i due programmi: spegnete, riaccendete e digitate tutto daccapo (purtroppo!).

Impartito il RUN al programma n.3, apparirà la richiesta di inserire un valore compreso tra 1 e 255, indicante un non meglio precisato "ritardo".

Per capire cosa significhi, rispondete con 1 e premete Return. Sullo schermo apparirà qualcosa di assai diverso dallo sprite rettangolare, come nel listato 1: quello che sembra un disturbo del video, è in realtà proprio lo sprite gestito dal programma n.1 (in puro Basic), ma che percorre lo schermo ad una tale velocità da risultare, per una serie di motivi che non stiamo qui a precisare,, illeggibile nella sua "vera" forma.

Premete ora la barra spaziatrice: riapparirà l'input precedente. Se in risposta date ora il valore massimo del ritardo possibile, cioè 255, ecco che lo sprite apparirà... in tutto il suo splendore. Per modificarne la velocità provate ad usare valori diversi di ritardo, premendo ogni volta lo spazio per tornare all'input.

Una bella differenza rispetto al Basic, non trovate?

Questa differenza, di per sè, e già sufficiente a giustificare un certo sforzo di apprendimento dell'Assembly.

Editing col Macro Assembler

Dopo i primi passi in Basic, vediamo ora di muovere i primi veri passi nella programmazione in Assembly. In altre parole, è giunto il momento di sedersi davanti al nostro C/64, con accanto il dischetto del Macro Assembler Commodore pronto per l'uso. Come infatti si è detto in precedenza su questa stessa rivista, la realizzazione di un programma in Assembly è umanamente(!) possibile solo adoperando programmi particolari (gli editor/assembler) che facilitano la vita al programmatore evitando principi di pazzia: tra questi il Macro Assembler è senz'altro uno dei più completi e versatili che si possano trovare in commercio.

```
1 REM ***                               ***
2 REM ***    L I S T A T O    2         ***
3 REM ***                               ***
4 :
10 FORI=49152TO49208:READA
13 B=B+A:POKEI,A:NEXT
15 IF B<>7984THENPRINT"ERRORE!":END
200 :
210 DATA 169,000,141,016,208,162,000
220 DATA 142,000,208,032,046,192,232
230 DATA 208,247,169,001,141,016,208
240 DATA 160,088,162,000,142,000,208
250 DATA 032,046,192,232,136,208,246
260 DATA 032,228,255,201,032,208,001
270 DATA 096,076,000,192,165,251,133
280 DATA 252,198,252,234,234,208,250
290 DATA 096
```

```
1 REM ***                            ***
2 REM ***    L I S T A T O    3      ***
3 REM ***                            ***
4 :
20 VC=53248:Y=150
30 FOR I=0 TO 62
40 POKE 832+I,255:REM -> DEF. SPRITE
50 NEXT
60 POKE 2040,13:REM ---> PUNTATORE
70 POKEVC+39,1:REM ---> COLORE
80 POKEVC+21,1:REM ---> ATTIVAZIONE
85 POKEVC+1,Y: REM ---> COORD. VERT.
90 REM----------------------------
100 PRINT CHR$(147)
110 INPUT "RITARDO (1-255)";W
120 IFW<1ORW>255THEN100
130 POKEVC+21,1:REM ---> ATTIVAZIONE
140 POKE251,W:SYS49152
150 POKEVC+21,0:REM -> DISATTIVAZIONE
160 GOTO100
```

In Assembly è sufficiente un piccolo errore per "bloccare" il C/64

Come farlo in Assembly

La prima fase della realizzazione di una routine, indipendentemente dal linguaggio adoperato, consiste, ovviamente, nella sua stesura.

A questo scopo, nel dischetto Macro Assembler è contenuto un file dal nome "editor64", che va caricato con la sintassi (,8,1) e attivato mediante il comando SYS 49152.

Dopo un messaggio di copyright, ricomparirà il solito READY, come se nulla fosse successo, ma in realtà l'editor è già in funzione. Questo significa che il computer continuerà a rispondere ai comandi impartiti in modo diretto, ma non è più in grado di eseguire un programma in Basic.

Inoltre, si hanno ora a disposizione anche altre risorse, ovviamente descritte nel manuale d'uso, che esamineremo al momento del loro utilizzo.

Se avete deciso di utilizzare il Macro Assembler, non rimane che iniziare a digitare quello che possiamo definire un vero e proprio listato, numerato come nel Basic, e che potete vedere riprodotto in queste pagine.

In effetti ci si trova davanti ad un cosiddetto "disassemblato" (disassembler = operazione inversa dell'assembler), ma con caratteristiche molto più chiare rispetto allo stesso tipo di output visibile con comuni monitor. A differenza di questi ultimi, inoltre, con l'editor del Macro Assembler l'uso della notazione esadecimale è facoltativa, in quanto lo stesso è in grado di "riconoscere" i numeri decimali grazie all'assenza del simbolo del dollaro ($), obbligatorio per valori espressi in esadecimale: si veda, nelle prime righe del disassemblato, come siano usate entrambe le notazioni per assegnare valori.

Oltre a questo, è possibile anche indicare operazioni aritmetiche nella specificazione degli operandi (per esempio VC+16 della riga 220).

Come potete notare, a differenza del Basic, ogni riga può contenere solo una istruzione, ma il trattamento delle stesse è identico. E' possibile quindi effettuare un LIST, inserire una nuova riga semplicemente digitandola con il numero voluto, e così via.

In più, esiste l'istruzione:

NUMBER xx,yy,zz

...che consente di rinumerare l'intero listato indicando con xx il vecchio numero di linea da cui partire, con yy il nuovo numero, e con zz l'incremento; in altre parole il Macro Assembler consente di rilocare un programma in linguaggio macchina.

Copiando il listato Macro assembler, potrà inoltre risultare utile il comando AUTO xx, che produce automaticamente il nuovo numero di linea, dopo averne digitata una, secondo l'incremento indicato da xx.

Nel redigere il programma, occorre comunque prestare una certa attenzione nell'uso degli spazi. Nell' editor, infatti, essi rivestono una notevole importanza, che però esamineremo in seguito. Se avete optato sin d'ora per l'utilizzo del Macro Assembler, trascriveteli così come sono disposti in figura, tenendo presente che il simbolo di punto e virgola (;) corrisponde in tutto e per tutto alla REM del Basic, per cui potete modificare (o eliminare) tutto ciò che segue questo carattere.

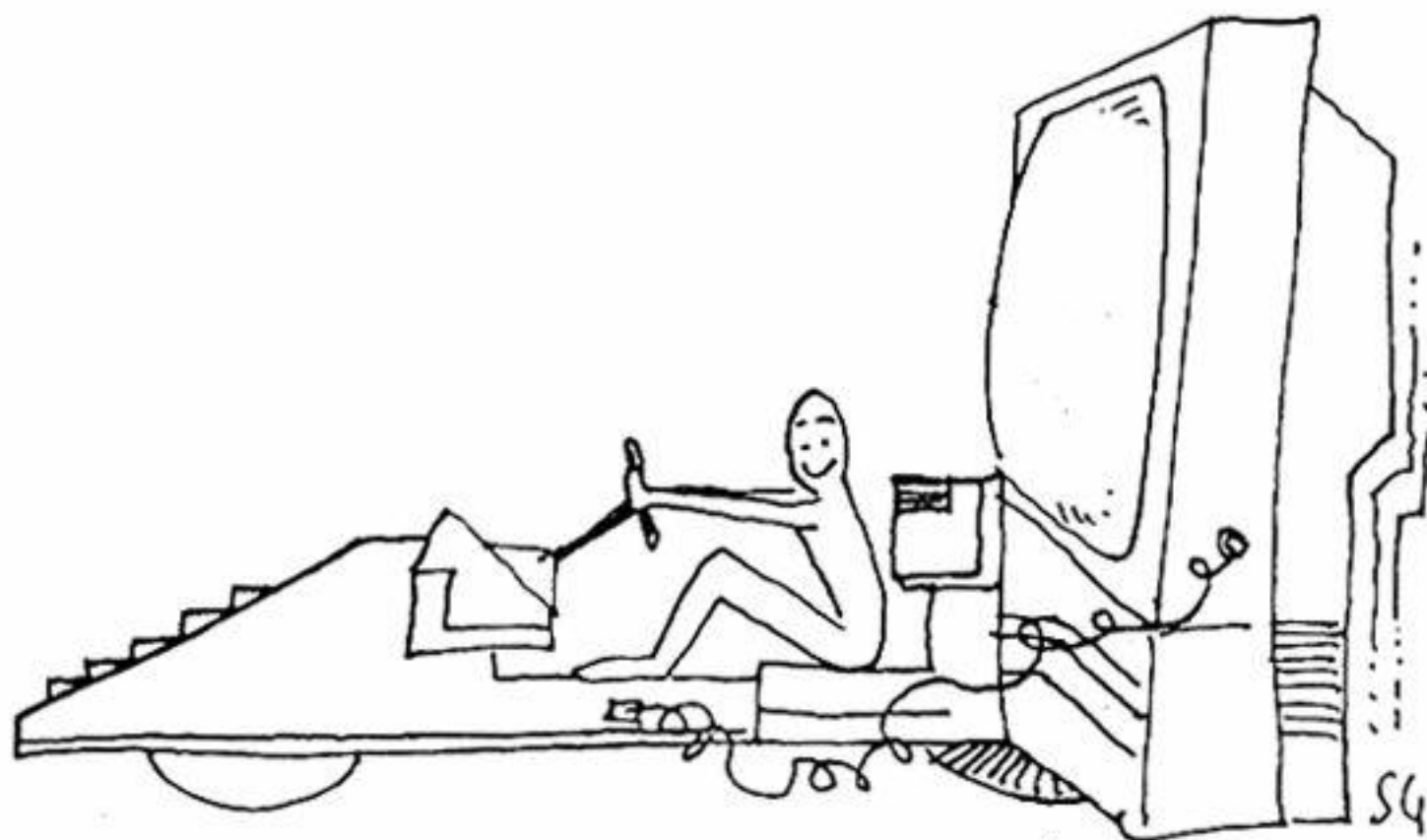
Ultimata la fase di stesura, si procede a salvare il tutto con il comando:

PUT "nome file"

...che creerà sul disco (non necessariamente lo stesso del Macro assembler) un file sequenziale (=file sorgente) col nome assegnatogli. Se si vuole ricaricare un file precedentemente salvato, l'istruzione relativa sarà:

GET "nome file"

Per giungere ad avere in memoria il codice l.m. saranno necessari ancora due passaggi, nei quali però tutto il lavoro verrà svol-

to dal computer. Il nostro intervento si limiterà, in sintesi, alle seguenti operazioni:

1) Uscire dall'editor tramite il comando KILL (oppure resettare il computer).

2) Caricare il programma "Assembler64" e lanciarlo con un normale RUN.

3) Rispondere alla prima domanda assegnando un nome al file che sarà generato dall'assembler (object file = file oggetto), e alla seconda Y (yes) o N (no) per indicare se volete una copia del disassemblato completo sulla stampante.

Alla successiva richiesta (Cross Reference?) per il momento rispondete N, poi date il nome col quale avete salvato il file prodotto con l'editor (il file sorgente). L'assembler svolgerà ora il suo compito, producendo sul dischetto un nuovo file sequenziale contenente il codice macchina, ma in formato ASCII.

4) Per utilizzare effettivamente la routine, non resta che caricare uno dei programmi loader, in questo caso "loloader" (proprio così, non è un errore di stampa), e dopo il RUN fornirgli il nome del file oggetto, rispondendo negativamente alla richiesta di un offset.

A questo punto avremo in memoria la routine vera e propria, che potremo osservare direttamente, modificare, o salvare nel suo formato l.m. tramite il monitor.

Date le enormi possibilità del Macro Assembler, una sua generica descrizione globale non può esimersi dal risultare un po' complicata, ma tutto risulterà sempre più chiaro a mano a mano che ne approfondiremo i particolari nei prossimi appuntamenti.

Il disassemblato

Se analizziamo il disassemblato Macro assembler della routine, ci si accorgerà che questa è del tutto simile alla parte inferiore del listato Basic n.1, con le istruzioni di Load e Store che sostituiscono le Poke del corrispettivo in Basic.

Alcuni dei codici Assembly presenti nella routine non vi sono ancora noti, per cui accontentatevi dei commenti posti alla loro destra. Tra tutti, comunque, sappiate che JSR indica un'istruzione di salto (come il GOSUB del Basic) all'indirizzo indicato dall'operando. Al posto di quest'ultimo, per semplificare il lavoro del programmatore, possono essere utilizzati particolari punti di riferimento chiamati "etichette" (= label), come "inizio", "ciclo1", eccetera: provvederà poi l'assembler a trasformarle in indirizzi.

Data l'abbondanza di note esplicative, il paragone tra il disassemblato ed il listato 1 è abbastanza evidente; l'unico punto forse meno chiaro può risultare il ciclo di ritardo delle linee da 560 in poi.

Questo è inserito nel programma Assembler come una subroutine, che viene chiamata ogni volta che si incrementa la posizione dello sprite sullo schermo. Il compito che svolge può essere paragonato ad un ciclo "vuoto" FOR...NEXT, in cui il "contatore" è costituito dal contenuto della locazione 252.

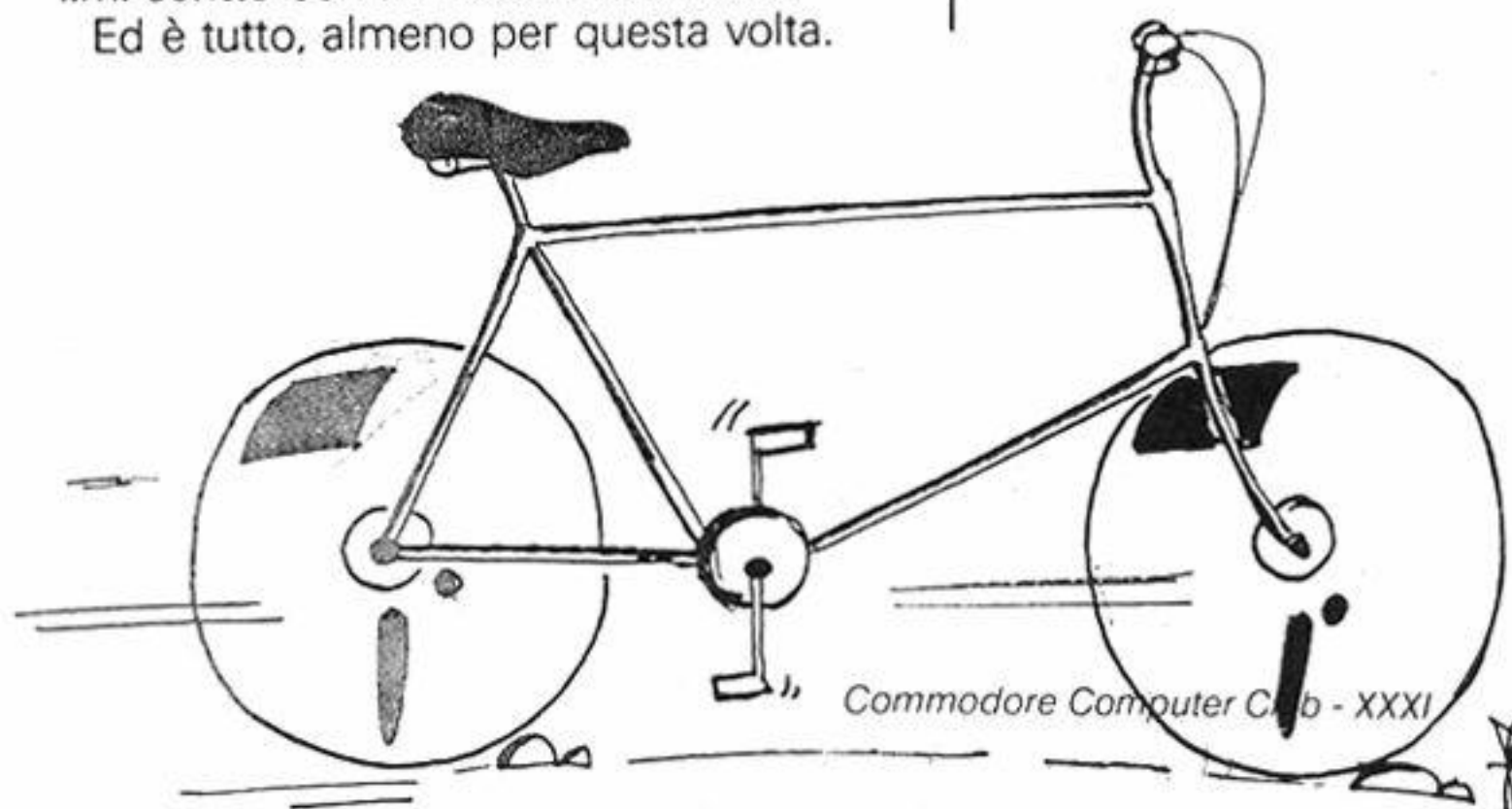
Questo valore (linee 560-570) viene prelevato dalla locazione 251, dove era stato depositato dalla POKE di riga 140 del listato 3, e poi semplicemente decrementato (DEC) di un'unità finchè non raggiunge il valore nullo.

Tutto questo semplicemente per... perdere un po' di tempo. L'istruzione DEC (della quale esiste anche il contrario: INC) è analoga alle già viste DEX e DEY, solo che agisce sulla memoria e non sui registri. Come si può rilevare anche dalle righe 330 - 380, questo tipo di codici vengono spesso usati per creare cicli iterativi, tenendo presente quanto segue: il contenuto di un registro (o di una locazione), se incrementato oltre il limite di 255 ricomincia da 0; viceversa, se decrementato al di sotto di 0, ricomincia da 255.

Ultima nota prima di concludere: il simbolo "*=" presente in riga 160 indica all'assembler da quale locazione il programma l.m. dovrà cominciare, informazione questa che è indispensabile per un suo corretto funzionamento e che consente, come già detto, di rilocare un qualsiasi programma l.m. scritto con il Macro Assembler.

Ed è tutto, almeno per questa volta.

Un disassemblato è complesso solo in apparenza

La velocità ottenibile in l.m. è talmente veloce che nemmeno il processore video è in grado di gestire correttamente lo sprite

```
100 ; movimento sprite
110 ;
120 vc       =53248
130 input    =$FB         ; dec.251
140 locmem   =$FC         ; dec.252
150 getin    =$FFE4       ; dec.65508
160 start    *=49152
170 ;
180 ;---------------------------------
190 ;******  posizione x =< 255  *****
200 ;---------------------------------
210 inizio   lda #0
220          sta vc+16   ;poke vc+16,0
230          ldx #0      ;x=0
240 ciclo1   stx vc      ;movimento
250          jsr wait    ;ciclo ritardo
260          inx         ;x=x+1
270          bne ciclo1  ;se x<>0 salta
280 ;---------------------------------
290 ;******  posizione x > 255  ******
300 ;---------------------------------
310          lda #1
320          sta vc+16   ;poke vc+16,1
330          ldy #88     ;y=88
340          ldx #0      ;x=0
350 ciclo2   stx vc      ;movimento
360          jsr wait    ;ciclo ritardo
370          inx         ;x=x+1
380          dey         ;y=y-1
390          bne ciclo2  ;se y<>0 salta
400 ;---------------------------------
410 ;******   spazio premuto?   ******
420 ;---------------------------------
430          jsr getin   ;GET carattere
440          cmp #32     ;e' lo spazio?
450          bne loop    ;se no, salta
460          rts         ;torna al basic
470 ;---------------------------------
480 ;****  ricomincia dall'inizio ****
490 ;---------------------------------
500 ;
510 loop     jmp inizio  ;salta
520 ;
530 ;---------------------------------
540 ;******  ciclo di ritardo   ******
550 ;---------------------------------
560 wait     lda input   ;carica W
570          sta locmem  ;lo pone in 252
580 ciclo3   dec locmem  ;W=W-1
590          nop         ;non operativo
600          nop         ;
610          bne ciclo3  ;se W<>0 salta
620          rts         ;return
```

UN SISTEMA PICCOLO PICCOLO

Il computer vi aiuta ad eliminare le colonne di un sistema totocalcio che hanno minime probabilità di vincita

di **Antonio Pastorelli**

Come promesso sul fascicolo speciale di "Commodore", dedicato interamente al Totocalcio, riprendiamo il discorso proponendo alcuni riduttori idonei per ampliare le capacità del programma Tot Reducer V 1.0.

Per non ripeterci, e per non annoiare il lettore con lunghi discorsi teorici, eviteremo di spiegare ancora una volta che cosa siano i sistemi ridotti; ci limiteremo quindi a fornire il listato da digitare, per ampliare il programma in questione, e le necessarie istruzioni per il corretto utilizzo.

Chi fosse interessato all'argomento potrà procurarsi il numero 18 della rivista "Commodore" od il numero 37 di "Commodore Computer Club", dove troverà articoli riguardanti i sistemi ridotti, corredati da relativi programmi.

I SISTEMI RIDOTTI

Nonostante sia già evidenziato nell'articolo, è bene sottolineare che il fascicolo "Commodore" speciale n.18, dedicato interamente al gioco del Totocalcio da sviluppare sul Commodore 64, comprende una cassetta contenente tutti i programmi presenti sul fascicolo stesso.

Speciale Totocalcio, infatti, è uno dei risultati dell'iniziativa "Software made in Italy" che si propone di offrire, agli utenti Commodore, software non protetto, trasferibile su nastro o disco e, soprattutto, ampiamente documentato per consentire a chiunque di comprenderne il funzionamento e, magari, apportare modifiche.

Il programma "Tot Reducer", di cui si parla nel presente articolo, è uno dei listati pubblicati su Commodore Speciale e risulta indispensabile per l'applicazione suggerita.

Il listato presente questo mese, quindi, non può funzionare se non inserito in coda a "Tot Reducer".

Nonostante sia possibile, ancora oggi, procurarsi in edicola lo Speciale Totocalcio (al prezzo irrisorio di L.8000), i lettori potranno richiederlo al nostro servizio arretrati (tel.02/84.67.34.8).

Nel listato che segue sono contenuti i dati per ridurre i seguenti sistemi:

- *8 doppie*
- *5 triple*
- *3 triple + 4 doppie*

La procedura da seguire, per l'implementazione dei riduttori sul programma "Tot Reducer", è la seguente:

- *Caricare in memoria il programma "Tot Reducer".*

- *Aumentare il numero di riduttori diponibili.*

Se possedete la versione originale di Tot Reducer, privo cioè di eventuali modifiche apportate per aumentare il numero di riduttori, digitate la seguente linea:

2920 Data 17: Rem numero riduttori previsti

Nel caso in cui abbiate già inserito vostri riduttori, anzichè il valore 17 (nella linea 2920) dovrete inserire il numero di riduttori disponibili in complesso; tale valore si determina facilmente sommando, al numero di riduttori del vostro programma, i tre riduttori pubblicati in queste pagine.

Se, ad esempio, avete aggiunto 5 riduttori alla versione originale, allora dovrete inserire, nella linea 2920, il numero 22: 14 iniziali + 5 aggiunti + 3 proposti questo mese = 22.

- *Inserire nel programma la descrizione dei nuovi riduttori.*

A tale scopo digitate la seguente linea:

3912 Data 8D,10000,5T,10430, 3T4D,10710

Nel caso in cui abbiate già utilizzato la linea 3912 per la descrizione di riduttori da voi implementati, potete aggiungere, in coda, quelli nuovi, o utilizzare un numero di linea maggiore, come indicato nell'articolo a corredo del fascicolo Commodore speciale.

10000, 10430 e 10710, sono infatti i numeri di linea dai quali iniziano i dati dei tre riduttori proposti; potete

```
Primo sistema
(8 doppie):
1X
 1
 X
X1
X2
 1
 1
21
X1
 1
2X
12
1X
Riduzione:

  1) 11XXX112X1211
  2) 11X1X11211X11
  3) 11XX211211221
  4) 11X12112X1X21
  5) 11XXX111X1X21
  6) 11X1X11111221
  7) 11XX211111X11
  8) 11X12111X1211
  9) X1XXX11211X21
 10) X1X1X112X1221
 11) X1XX2112X1X11
 12) X1X1211211211
 13) X1XXX11111211
 14) X1X1X111X1X11
 15) X1XX2111X1221
 16) X1X1211111X21
 17) 11XXX112X121X
 18) 11X1X11211X1X
 19) 11XX21121122X
 20) 11X12112X1X2X
 21) 11XXX111X1X2X
 22) 11X1X1111122X
 23) 11XX211111X1X
 24) 11X12111X121X
 25) X1XXX11211X2X
 26) X1X1X112X122X
 27) X1XX2112X1X1X
 28) X1X121121121X
 29) X1XXX1111121X
 30) X1X1X111X1X1X
 31) X1XX2111X122X
 32) X1X1211111X2X
```

```
Secondo sistema
(5 triple):

  1
  1
  X
1X2
  1
X12
X21
  1
  1
  X
2X1
  X
12X

Riduzione:

  1) 11X11xx11x2x1
  2) 11xx11211x2x1
  3) 11x212111x2x1
  4) 11x111111x2x1
  5) 11xx12x11xxx1
  6) 11x21x211xxx1
  7) 11x112211x1x1
  8) 11xx1x111x1x1
  9) 11x211x11x1x1
 10) 11x11xx11x2x2
 11) 11xx11211x2x2
 12) 11x212111x2x2
 13) 11x111111x2x2
 14) 11xx12x11xxx2
 15) 11x21x211xxx2
 16) 11x112211x1x2
 17) 11xx1x111x1x2
 18) 11x211x11x1x2
 19) 11x11xx11x2xx
 20) 11xx11211x2xx
 21) 11x212111x2xx
 22) 11x111111x2xx
 23) 11xx12x11xxxx
 24) 11x21x211xxxx
 25) 11x112211x1xx
 26) 11xx1x111x1xx
 27) 11x211x11x1xx
```

```
Terzo sistema (3 triple + 4 doppie):

   1
   1
  1X
  X1
  12
   X
   X
 1X2
   1
 X21
 12X
  1X
   2

Riduzione:

 1) 111x1xxx11112
 2) 111x1xx21x212
 3) 111x1xx212x12
 4) 111x1xx11xx12
 5) 111x1xx112212
 6) 11xx1xxx11112
 7) 11xx1xx21x212
 8) 11xx1xx212x12
 9) 11xx1xx11xx12
10) 11xx1xx112212
11) 11111xxx11112
12) 11111xx21x212
13) 11111xx212x12
14) 11111xx11xx12
15) 11111xx112212
16) 11x11xxx11112
17) 11x11xx21x212
18) 11x11xx212x12
19) 11x11xx11xx12
20) 11x11xx112212
21) 111x2xxx11112
22) 111x2xx21x212
23) 111x2xx212x12
24) 111x2xx11xx12
25) 111x2xx112212
26) 11xx2xxx11112
27) 11xx2xx21x212
28) 11xx2xx212x12
29) 11xx2xx11xx12
30) 11xx2xx112212
31) 11112xxx11112
32) 11112xx21x212
33) 11112xx212x12
34) 11112xx11xx12
35) 11112xx112212
36) 11x12xxx11112
37) 11x12xx21x212
38) 11x12xx212x12
39) 11x12xx11xx12
40) 11x12xx112212
41) 111x1xxx111x2
42) 111x1xx21x2x2
43) 111x1xx212xx2
44) 111x1xx11xxx2
45) 111x1xx1122x2
46) 11xx1xxx111x2
47) 11xx1xx21x2x2
48) 11xx1xx212xx2
```

inserire i dati dei riduttori utilizzando una diversa numerazione, ma in tal caso dovrete sostituire, in linea 3912, i valori 10000, 10430 e 10710, con quelli da voi utilizzati.

Se, ad esempio, utilizzate le linee da 20000 in poi, la linea 3912 andrà digitata nel modo seguente:

3912 Data 8D,20000,5T,20430, 3T4D,20710

Ed ora non resta che digitare pazientemente il listato.

Alcune precauzioni

Per evitare di manomettere irrimediabilmente, a causa di errori di digitazione, il listato originale di Tot Reducer, è importante far attenzione a non registrare il programma modificato sostituendolo a quello originale.

Chi utilizza il drive non dovrebbe avere questo problema, in quanto non può registrare su un disco due programmi con lo stesso nome; in ogni caso è bene salvare a parte la versione modificata, per testarla prima dell'utilizzo.

E' quindi consigliabile conservare sempre una copia di sicurezza del programma originale.

In caso di errore di trascrizione, purtroppo, il computer non può segnalare errori di alcun genere, ma provvede, sempre e comunque, a visualizzare un certo numero di colonne apparentemente "normali".

Per testare il funzionamento di Tot Reducer modificato, pertanto, vi proponiamo tre sistemi che utilizzano i riduttori appena inseriti. Potrete quindi confrontare i risultati con le colonne che indichiamo per ciascun sistema.

```
9000 REM RIDUTTORI AGGIUN
9010 REM TOT REDUCER V1.0
9020 REM     PUBBLICATO. SU
9030 REM    "COMMODORE N.18"
9040 REM
9050 REM BY A. PASTORELLI
9060 :
10000 DATA 32:REM COL.PER 8D
10010 DATA 1,1,1,1,1,1,1,1
10020 DATA 1,2,1,1,2,2,1,1
10030 DATA 1,1,2,1,2,1,2,1
10040 DATA 1,2,2,1,1,2,2,1
10050 DATA 1,1,1,2,1,2,2,1
10060 DATA 1,2,1,2,2,1,2,1
10070 DATA 1,1,2,2,2,2,1,1
10080 DATA 1,2,2,2,1,1,1,1
10090 DATA 2,1,1,1,2,2,2,1
10100 DATA 2,2,1,1,1,1,2,1
10110 DATA 2,1,2,1,1,2,1,1
10120 DATA 2,2,2,1,2,1,1,1
10230 DATA 2,1,1,2,2,1,1,1
10240 DATA 2,2,1,2,1,2,1,1
10250 DATA 2,1,2,2,1,1,2,1
10260 DATA 2,2,2,2,2,2,2,1
10270 DATA 1,1,1,1,1,1,1,2
10280 DATA 1,2,1,1,2,2,1,2
10290 DATA 1,1,2,1,2,1,2,2
10300 DATA 1,2,2,1,1,2,2,2
10310 DATA 1,1,1,2,1,2,2,2
10320 DATA 1,2,1,2,2,1,2,2
10330 DATA 1,1,2,2,2,2,1,2
10340 DATA 1,2,2,2,1,1,1,2
10350 DATA 2,1,1,1,2,2,2,2
10360 DATA 2,2,1,1,1,1,2,2
10370 DATA 2,1,2,1,1,2,1,2
10380 DATA 2,2,2,1,2,1,1,2
10390 DATA 2,1,1,2,2,1,1,2
10400 DATA 2,2,1,2,1,2,1,2
10410 DATA 2,1,2,2,1,1,2,2
10420 DATA 2,2,2,2,2,2,2,2
10425 :
10430 DATA 27:REM COLON.ST
10440 DATA 1,1,1,1,1
10450 DATA 2,2,2,1,1
10460 DATA 3,3,3,1,1
10470 DATA 1,2,3,1,1
10480 DATA 2,3,1,2,1
10490 DATA 3,1,2,2,1
10500 DATA 1,3,2,3,1
10510 DATA 2,1,3,3,1
10520 DATA 3,2,1,3,1
10530 DATA 1,1,1,1,2
10540 DATA 2,2,2,1,2
10550 DATA 3,3,3,1,2
10560 DATA 1,2,3,1,2
10570 DATA 2,3,1,2,2
10580 DATA 3,1,2,2,2
10590 DATA 1,3,2,3,2
10600 DATA 2,1,3,3,2
10610 DATA 3,2,1,3,2
10620 DATA 1,1,1,1,3
10630 DATA 2,2,2,1,3
10640 DATA 3,3,3,1,3
10650 DATA 1,2,3,1,3
10660 DATA 2,3,1,2,3
10670 DATA 3,1,2,2,3
10680 DATA 1,3,2,3,3
10690 DATA 2,1,3,3,3
10700 DATA 3,2,1,3,3
10705 :
10710 DATA 48:REM    3T4D
10720 DATA 2,3,1,1,1,1,1
10730 DATA 3,1,2,1,1,1,1
10740 DATA 3,2,3,1,1,1,1
10750 DATA 1,1,3,1,1,1,1
10760 DATA 1,2,2,1,1,1,1
10770 DATA 2,3,1,2,1,1,1
10780 DATA 3,1,2,2,1,1,1
10790 DATA 3,2,3,2,1,1,1
10800 DATA 1,1,3,2,1,1,1
10810 DATA 1,2,2,2,1,1,1
10820 DATA 2,3,1,1,2,1,1
10830 DATA 3,1,2,1,2,1,1
10840 DATA 3,2,3,1,2,1,1
10850 DATA 1,1,3,1,2,1,1
10860 DATA 1,2,2,1,2,1,1
10870 DATA 2,3,1,2,2,1,1
10880 DATA 3,1,2,2,2,1,1
10890 DATA 3,2,3,2,2,1,1
10900 DATA 1,1,3,2,2,1,1
10910 DATA 1,2,2,2,2,1,1
10920 DATA 2,3,1,1,1,2,1
10930 DATA 3,1,2,1,1,2,1
10940 DATA 3,2,3,1,1,2,1
10950 DATA 1,1,3,1,1,2,1
10960 DATA 1,2,2,1,1,2,1
10970 DATA 2,3,1,2,1,2,1
10980 DATA 3,1,2,2,1,2,1
10990 DATA 3,2,3,2,1,2,1
11000 DATA 1,1,3,2,1,2,1
11010 DATA 1,2,2,2,1,2,1
11020 DATA 2,3,1,1,2,2,1
11030 DATA 3,1,2,1,2,2,1
11040 DATA 3,2,3,1,2,2,1
11050 DATA 1,1,3,1,2,2,1
11060 DATA 1,2,2,1,2,2,1
11070 DATA 2,3,1,2,2,2,1
11080 DATA 3,1,2,2,2,2,1
11090 DATA 3,2,3,2,2,2,1
11100 DATA 1,1,3,2,2,2,1
11110 DATA 1,2,2,2,2,2,1
11120 DATA 2,3,1,1,1,1,2
11130 DATA 3,1,2,1,1,1,2
11140 DATA 3,2,3,1,1,1,2
11150 DATA 1,1,3,1,1,1,2
11160 DATA 1,2,2,1,1,1,2
11170 DATA 2,3,1,2,1,1,2
11180 DATA 3,1,2,2,1,1,2
11190 DATA 3,2,3,2,1,1,2
```

GLI OPERATORI DEL LINGUAGGIO C

Continua la chiacchierata per avvicinarso gradualmente al linguaggio degli anni 2000

di **Luigi Callegari**

Ci mancano ancora alcuni elementi base del linguaggio C prima di essere in grado di scrivere semplici programmini. Dopo i tipi di dati, questa volta parleremo degli operatori, ovvero delle operazioni logiche e matematiche possibili.

Nessun linguaggio, a parte lo APL, ha più operatori del C, quindi l'argomento richiede una trattazione "succosa".

Abbiamo trattato, nella precedente "chiacchierata" (C.C.C. n.51), di alcuni tipi di dati del C; gli elementi base sono i numeri (interi od in virgola mobile) i caratteri ed i puntatori. Questi elementi, nel corpo di un programma, debbono essere manipolati per raggiungere risultati. Ciò si ottiene proprio con gli "operatori", in grado di ottenere un risultato partendo da uno o due operandi.

Gli operatori e gli operandi formano nel programma delle espressioni logiche o matematiche. Ovviamente, vi sono delle regole rigide che determinano il modo con cui si ottengono dei risultati partendo da espressioni. Ricordiamo che un'operazione matematica del tipo...

1 + 2 * 3

...ha come risultato 7. Però, se proviamo ad eseguire il calcolo con una economica (non scientifica-algebrica) calcolatrice tascabile da poche lire, avremo come risultato 9.

Qui si nota che il calcolatore tascabile segue regole di aggregazione dei tre operandi (1, 2, 3) e dei due operatori matematici (+ e *) in modo diverso da quanto, invece, usa un C/64 e la comune algebra. Infatti la macchinetta esegue il calcolo come se avessimo digitato...

(1 + 2) * 3

...svolgendo cioè i calcoli nell'ordine in cui sono scritti, mentre un calcolatore più evoluto valuta correttamente l'espressione:

1 + (2 * 3)

Esistono dunque delle "precedenze" tra gli operatori. Nel nostro caso, l'operatore di moltiplicazione va

SCHEDA TECNICA

Articolo didattico sul linguaggio "C"

Hardware richiesto: Amiga (qualsiasi configurazione) e, nei casi più semplici, C/64 (oppure C/128) e disk drive

Richiede la disponibilità del linguaggio "C"

Consigliato agli esperti

considerato prima di quello di somma (e dopo quello di potenza, se vi fosse). Il C, per i calcoli algebrici, conosce ovviamente le giuste priorità e fornisce i corretti risultati. Spesso, però, si usano espressioni con operatori logici o matematici (che vedremo oltre) in grado di influenzare le priorità e di falsare i risultati.

Comunque, in C vi sono le parentesi tonde, che permettono di specificare che una certa espressione deve essere valutata indipendentemente dalle precedenti o dalle successive e "forzando" dunque a calcolarla nel modo che noi stabiliamo. Ciò è possibile in (quasi) tutti i linguaggi, ma in C, ricordatelo, si è costretti ad usare le parentesi tonde più frequentemente del solito, data la sua sinteticità ed il numero di operatori.

Un operatore in C, materialmente, è costituito quasi sempre da uno o più simboli grafici consecutivi, a differenza di altri linguaggi dove si usano spesso parole.

Praticamente tutti i simboli grafici (non CBM-ASCII, ma ASCII standard) sono significativi e ciò provoca problemi in elaboratori come il C/64 dove, ad esempio, mancano le parentesi graffe direttamente sulla tastiera. Chi programma con Amiga sarà invece leggermente avvantaggiato se possiede una tastiera americana invece di quella nazionalizzata.

Operatori aritmetici

In C è possibile eseguire somme (+), sottrazioni (-), moltiplicazioni (*), divisioni (/). Gli operatori delle quattro operazioni sono ovviamente binari, cioè usano due operandi per produrre un risultato.

Inoltre è disponibile il modulo (%), noto ai programmatori AmigaBASIC, che fornisce il resto di una divisione intera. Ad esempio l'espressione...

```
7 % 3
```

...ha come risultato uno (1), in quanto il tre sta nel sette due volte con il resto di uno, come si studiava sui cari banchi delle elementari.

Non esiste l'operatore di elevamento a potenza. Ciò non significa che sia impossibile calcolare potenze e radici ennesime, ma che si dovrà ricorrere ad una "funzione" di libreria.

Qualcuno si chiederà come è "fatto" un operatore unario, che parte da un operando per dare un risultato. Ad esempio, facendo precedere un numero od un nome di variabile dal simbolo di meno (-) si applica un operatore unario di negazione algebrica. Anche in Basic si scrive:

```
v= -5: f= -v
```

Si noti che in C si userebbe il punto e virgola (;) invece del doppio punto (:) per separare le due istruzioni.

Oltre a questo operatore, ve ne sono altri quattro specifici del C, molto potenti, specie quando sono applicati agli indici di matrici ed ai puntatori. Si tratta degli operatori di auto-incremento e di auto-decremento. I simboli grafici sono costituiti, rispettivamente, da due simboli di somma (++) e da due di differenza (--) senza spazi intermedi. A seconda della loro posizione, rispetto ad un operando "v", si suddividono in:

operatore di auto-pre-incremento: ++v
operatore di auto-post-incremento: v++
operatore di auto-pre-decremento: --v
operatore di auto-post-decremento: v--

Scrivere, ad esempio...

```
j--
```

...equivale a scrivere una tipica linea Basic o C:

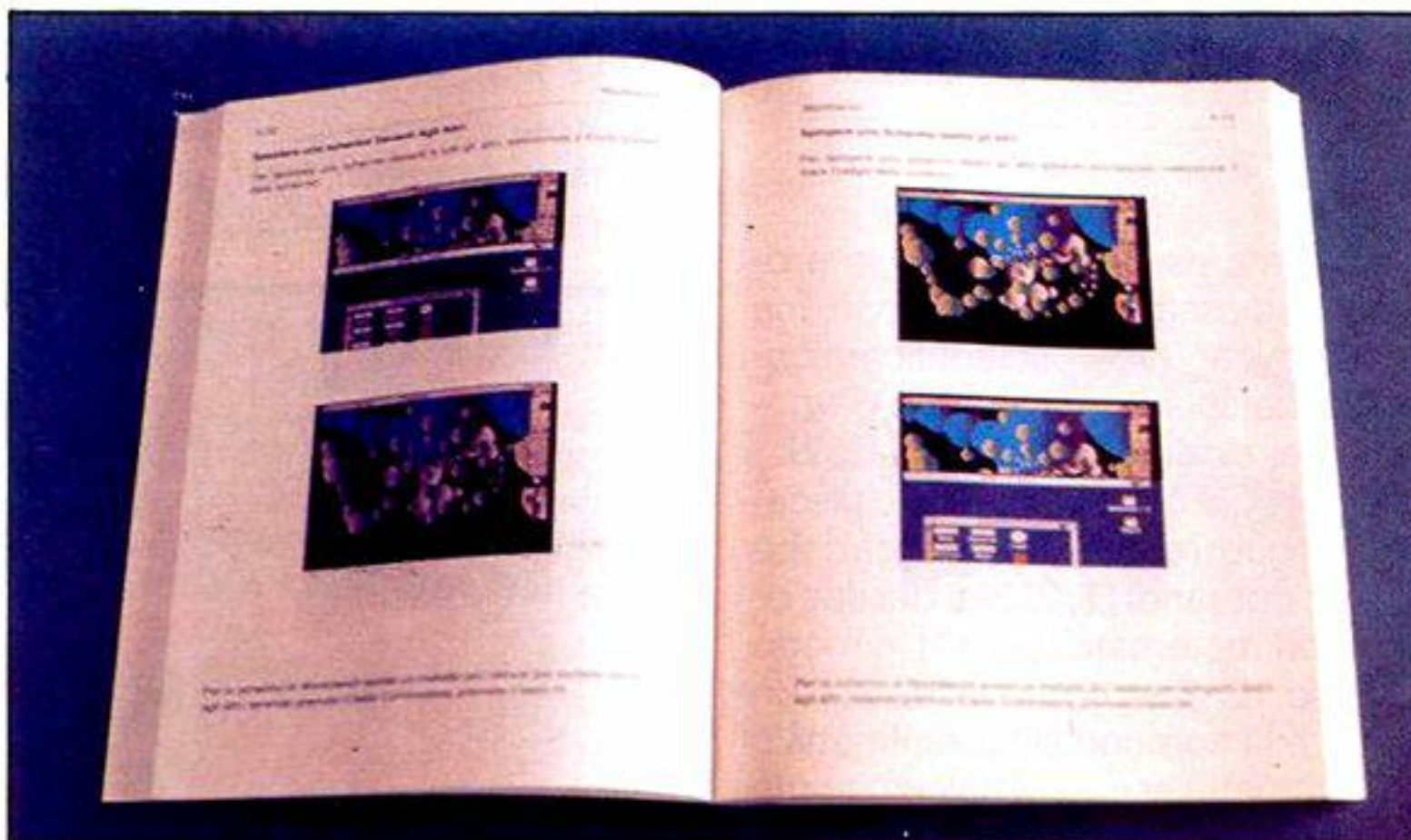
```
j=j - 1
```

Qui incominciamo ad intuire perchè tutti dicono che il C è un linguaggio sintetico. Si noti che i vantaggi permessi da simili operatori sarà afferrata pienamente quando si scriveranno programmi "veri". Comunque già ora è possibie notare che una espressione come...

```
++gabry[5 * 7 - index--]
```

...equivale all'espressione Basic (e C, volendo):

```
gabry[5*7-index]= gabry[5*7 - index]+1
index = index - 1
```

In "un colpo solo", usando gli operatori di auto-incremento, con C abbiamo detto che vogliamo incrementare il contenuto di una matrice, con indice calcolato tramite una espressione, e che una variabile dell'espressione di calcolo dell'indice deve essere poi decrementata (grazie all'operatore di auto-post-decremento).

Ciò, oltre ad evitare affaticamento dei polpastrelli delle dita, permette di fare eseguire il calcolo dell'indice una sola volta, invece di due (col secondo costrutto), dunque con notevole aumento nella velocità di esecuzione.

Si noti che i prefissi (++v, --v) indicano che l'incremento deve essere eseguito prima del calcolo della espressione globale, i postfissi (v--, v++) dopo.

Ciò significa che in C le seguenti linee...

```
i=5;  j=6;
k = i++ + --j;
```

...comporteranno che in uscita le variabili avranno i seguenti contenuti:

```
i = 6
j = 5
k = 5
```

in quanto k calcola effettivamente (5 + 5), operando subito il decremento di j (6-1 = 5) e posticipando il calcolo della intera espressione l'incremento della variabile i.

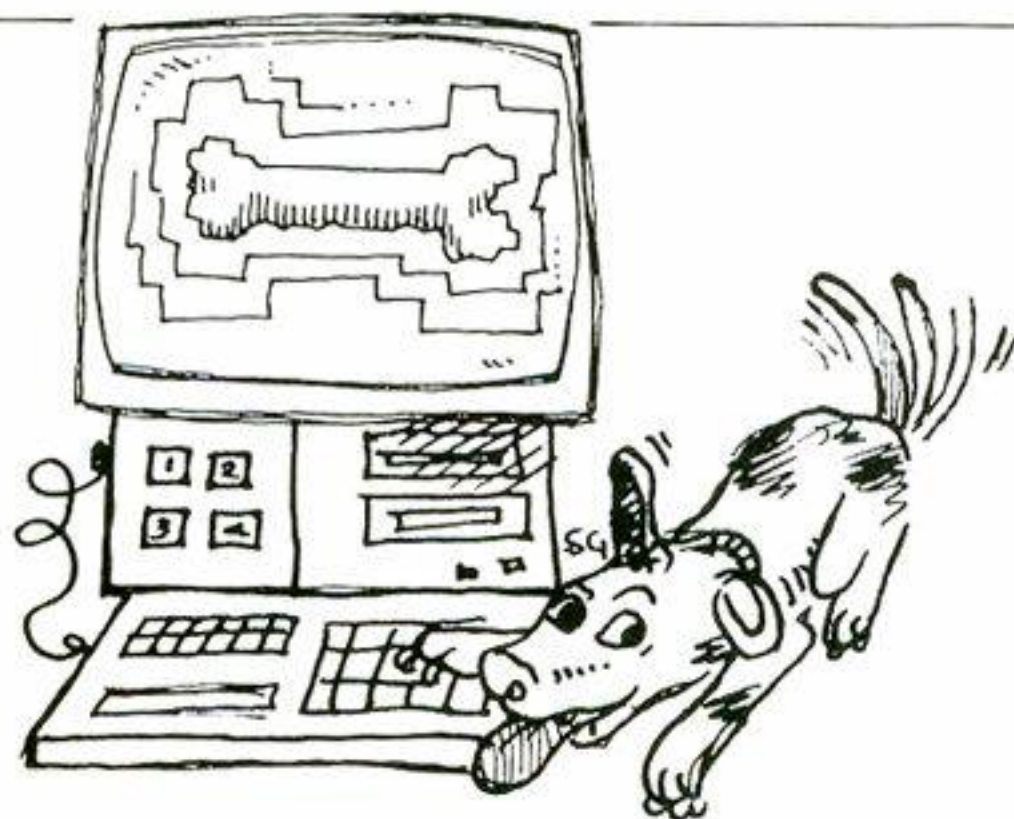
Chiariamo subito una cosa: vi sono operatori "complessi", che vedremo in futuro, che soverchiano anche quelli di autoincremento ed autodecremento. Di ciò bisognerà tenere conto per scrivere codici corretti, comunque per ora il discorso del calcolo "prima" e "dopo" nell'espressione globale può essere ritenuto corretto.

Operatori di calcolo ed assegnazione

Andiamo avanti a dimostrare la sinteticità di C. E' possibile anche semplificare espressioni di tipo (Basic):

```
a = a + b
c = c / e
```

Anche in tali casi un Basic perde tempo a calcolare due volte il valore di una stessa variabile (a, c). In C è

```
ALCUNI OPERATORI DEL LINGUAGGIO C

OPERATORI MATEMATICI

-           Negazione
++          Autoincremento
--          Autodecremento
+           Somma algebrica
-           Sottrazione
*           Moltiplicazione
/           Divisione
%           Modulo (resto)

OPERATORI LOGICI

!           Negazione logica
&&          AND logico
||          OR  logico

OPERATORI SUI BIT

~           Complemento
&           AND logico
|           OR  logico
^           XOR logico
<<          Scorrimento a sinistra
>>          Scorrimento a destra

OPERATORI DI CONFRONTO

==          Uguaglianza
!=          Diverso
>           Maggiore
<           Minore
>=          Maggiore od uguale
<=          Minore od uguale

OPERATORI DI CALCOLO ED ASSEGNAZIONE

=           Assegnazione
+=          Somma ed assegnazione
-=          Sottrazione ed assegnazione
*=          Moltiplica ed assegnazione
/=          Divisione ed assegnazione
%=          Modulo ed assegnazione
&=          AND logico sui bit ed assegnazione
|=          OR logico sui bit ed assegnazione
^=          XOR logico sui bit ed assegnazione
<<=         Scorrimento a sinistra ed assegnazione
>>=         Scorrimento a destra ed assegnazione
```

Alcune precisazioni per il C/64

Qualcuno avrà notato, leggendo i precedenti articoli sul C, che citiamo quasi sempre il Basic del C/64 ed il C (Lattice) di Amiga. Questo non perchè disdegniamo la vecchia scatola ad otto bit (con la quale scriviamo e lavoriamo correntemente) ma perchè abbiamo notato alcune cose e siamo realistici.

Innanzitutto conosciamo due versioni del C anche per il C/64 - C/128 ma non ci risulta siano ufficialmente importate; nel "mercato parallelo" i manuali, ASSOLUTAMENTE VITALI per usare questo linguaggio, sono praticamente inesistenti, anche perchè sono dei mattoni di carta e fotocopiarli è assai dispendioso e poco funzionale quando si commercia programmi piratati a mille lire al kilogrammo, come si fa ormai in Italia.

Inoltre, le versioni a noi note sono ben lontane dallo standard di Kernighan e Ritchie; una delle due versioni non accetta molti parametri tipici di printf, quindi praticamente inutili anche per chi voglia studiare un poco sul serio il linguaggio, con l'economico C/64.

Provando le versioni C per C/64 abbiamo notato che il sistema C/64 e 1541 per compilare programmi richiede una quantità di tempo incredibile, a causa della propria intrinseca "tartarugosità". Ricordiamo che compilare un programma C richiede consultazione di molti file (leggi, accessi al disco), tanto più frequenti quanto minore è la memoria centrale (e meno di 35K sono pochini). A quanto ci è dato di sapere, inoltre, non esistono "librerie" specializzate, come per Amiga, che permettano di sfruttare appieno le caratteristiche hardware della macchina (leggi, grafica e suono) come si deve.

Invece, con Amiga, un Lattice C (standard K&E e quasi-ANSI), importato dalla Commodore Italia e distribuito a circa novantamila lire, 512K ed il drive interno che trasferisce 25000 bit al secondo è già possibile fare cose quasi serie e comunque di provare in pratica tutto ciò che si trova sui libri di testo, anche se per un professionista od un dilettante evoluto ci vuole (come dicono sul manuale Lattice stesso) preferibilmente un hard-disk, due mega di memoria centrale ed una bella stampante.

In tal modo, infatti, diventa possibile per tutti sognare di scrivere videogiochi come Marble Madness o The Pawn (in puro C) usando librerie (fornite) e documentazione sul Kernel di Amiga, disponibile presso qualunque buona libreria scientifica internazionale (in stretto slang).

Comunque, per i possessori del solo C/64 + 1541, cercheremo di tenerci il più possibile sul "generico" nelle nostre pagine, aiutati dalla portatilità del C, anche se gli agganci pratici saranno qualche volta specifici per Amiga C.

possibile evitarlo. Infatti, quando si ha un costrutto del tipo...

(var) = (var) (operatore) (operando)

...si può scrivere semplicemente...

(var) (operatore)= (operando)

Ovvero, nei due casi prima citati:

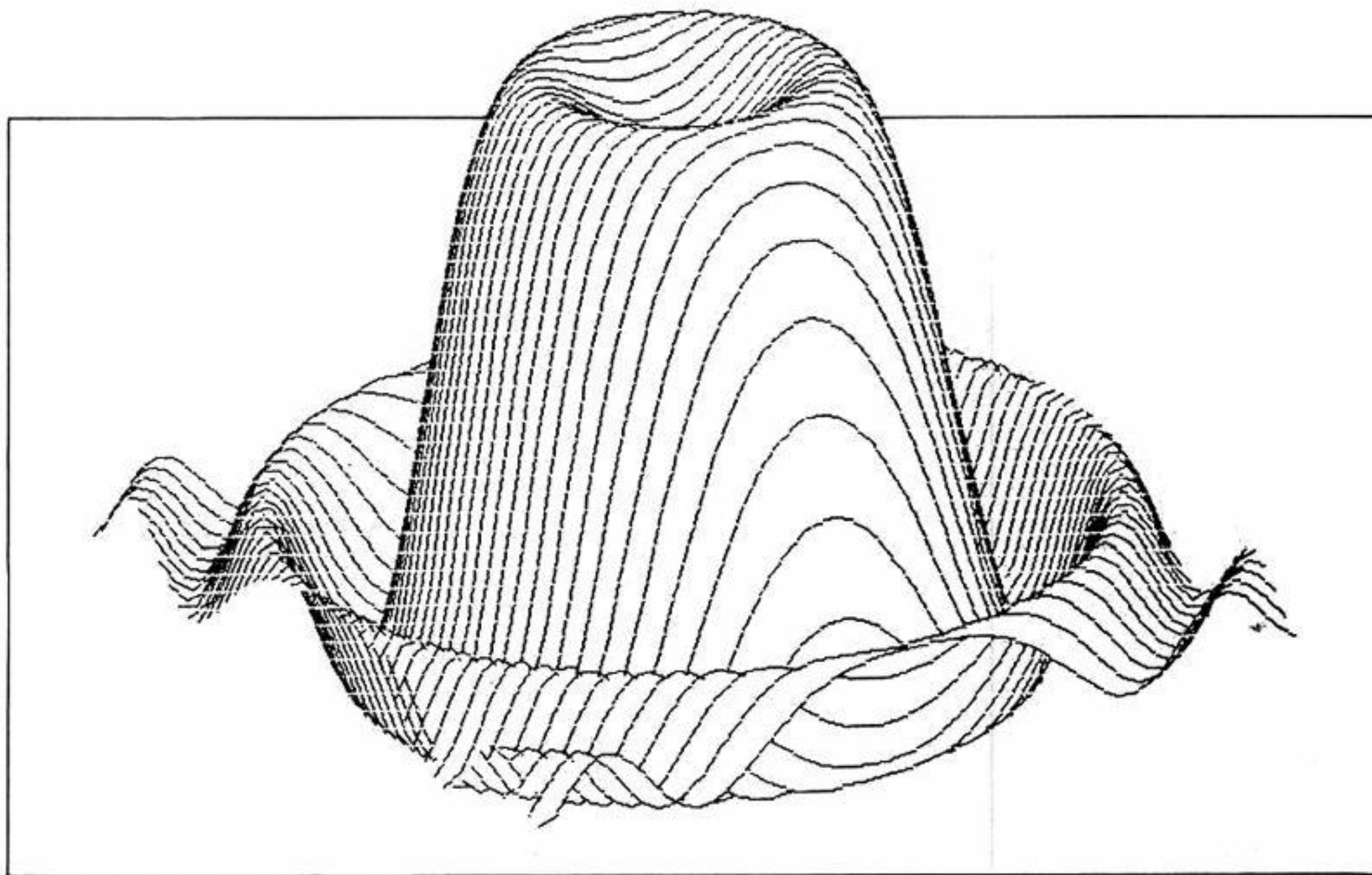
```
a += b
c /= e
```

Ancora, si risparmia spazio e si aiuta il compilatore a scrivere un codice più efficiente, in quanto si impone di cercare e leggere una variabile di meno, evitando la ripetizione della variabile di assegnazione.

In C sono ammessi dunque: +=, -=, /=, *=, %=

Le espressioni "tipo-basic" e quelle che sfruttano operatori di autoincremento, autodecremento o gli operatori di calcolo ed assegnazione sono perfettamente equivalenti, ma le seconde, come abbiamo spiegato, sono più efficienti.

Operatori sui bit

In C è possibile manipolare direttamenti dei Bit (ricordiamo che è stato sviluppato per scrivere sistemi operativi. Suoi figlioletti sono UNIX, MS-DOS e AmigaDOS). Chi ha programmato in Basic sul C/64 (o, almeno, ne ha letto il manuale) saprà che cosa significa eseguire lo AND e lo OR di operandi. In C si dispone anche di operatori che permettono direttamente di eseguire lo XOR ed il NOT (unario), nonchè di fare scorrere (Shift) lateralmente byte (8 bit) o word (16 bit). Siccome il loro uso è ristretto, quasi sempre, a quando si sta lavorando a stretto contatto con le routines del sistema operativo, non ci dilungheremo a descriverle ed ad esemplificarle. Chi deve usarle sa che ci sono e crediamo sia già abbastanza istruito da sapere che risultati producono.

RASSEGNA DI EFFETTI GRAFICI

Da questo numero di C.C.C. inizia la nuova raccolta di procedure software da utilizzare nei propri listati

di **Alessandro de Simone**

Sul numero scorso avevamo presentato l'ultima (si fa per dire) routine dell'enciclopedia Basic.

La raccolta di sottoprogrammi, tuttavia, era quasi esclusivamente dedicata ai possessori di C/64 perchè, al momento della prima proposta (ottbre 85), il C/128 ed il C/16 non avevano ancora raggiunto la diffusione odierna.

Con il passare del tempo, inoltre, il C/64 si è arricchito di vari package che gli hanno permesso di superare facilmente l'obsolescenza in cui rischiava di cadere, soprattutto per la carenza di specifiche istruzioni grafiche e sonore.

A partire da questo numero presenteremo numerose routine che permetteranno di realizzare una raccolta di procedure immediatamente utilizzabili per scopi specifici e che, soprattutto, rappresenteranno un'opportuna base di partenza per sviluppare software più complesso.

Le routine Basic pubblicate potranno essere utilizzate con i "piccoli" computer Commodore: C/128, C/16, Plus/4 e C/64; quest'ultimo dovrà esser dotato del linguaggio interprete "Gw-Basic".

La sintassi di numerosi comandi risulta, infatti, praticamente identica nel caso di C/128, C/16 e Plus/4. Per il C/64 (purchè dotato, lo ripetiamo, di Gw-Basic) è sufficiente apportare qualche modifica ai listati per ottenere le stesse elaborazioni.

Per evitare ripetizioni, quindi, verranno presentate, volta per volta, solo i programmi relativi ad un tipo di macchina e la corrispondente "traduzione" (limitata ad un solo listato), per l'altra categoria.

Il lettore, basandosi sulla traduzione riportata, sarà quindi facilitato nell'apportare le sostituzioni del caso.

Anche i possessori di altri Tool, tra cui le routine di Toma ed il Simon's Basic, non troveranno eccessive difficoltà nell'adattamento dei listati pubblicati.

I lettori chè volessero inviare le loro proposte, tengano presente che verranno presi in considerazione esclusivamente listati inviati su disco, corredati da una breve spiegazione sul funzionamento e sulle caratteristiche dell'utility inviata. Anche l'articolo, quindi, deve necessariamente essere presente su dischetto.

Assi cartesiani

Spesso, specialmente avendo a che fare con programmi matematici, si ha la necessità di tracciare i due

assi cartesiani, magari arricchiti di un sistema di misurazione che consenta di individuare la posizione di punti, segmenti, curve e così via.

Questa routine, che proponiamo sia per il C/16 (e C/128) che per C/64 (dotato di Gw-Basic) è in grado di tracciare rapidamente gli assi di riferimento richiesti.

Alla domanda "Asse X?" di dovrà rispondere con un valore, compreso tra 0 e 199, che rappresenterà il punto dell'asse Y preso come riferimento per il disegno dell'asse X; all'analoga risposta sull'asse Y si ripsonderà indicando il punto dell'asse X desiderato.

La lunghezza del segmento, cui si riferisce la domanda di riga 190, rappresenterà la lunghezza dei trattini, perpendicolari agli assi, che rappresenteranno un primo sistema di riferimento; digitando un valore positivo i trattini verranno riportati sulla destra (asse Y) e rivolti in basso (asse X). Si otterrà l'orientamento opposto digitando valori negativi.

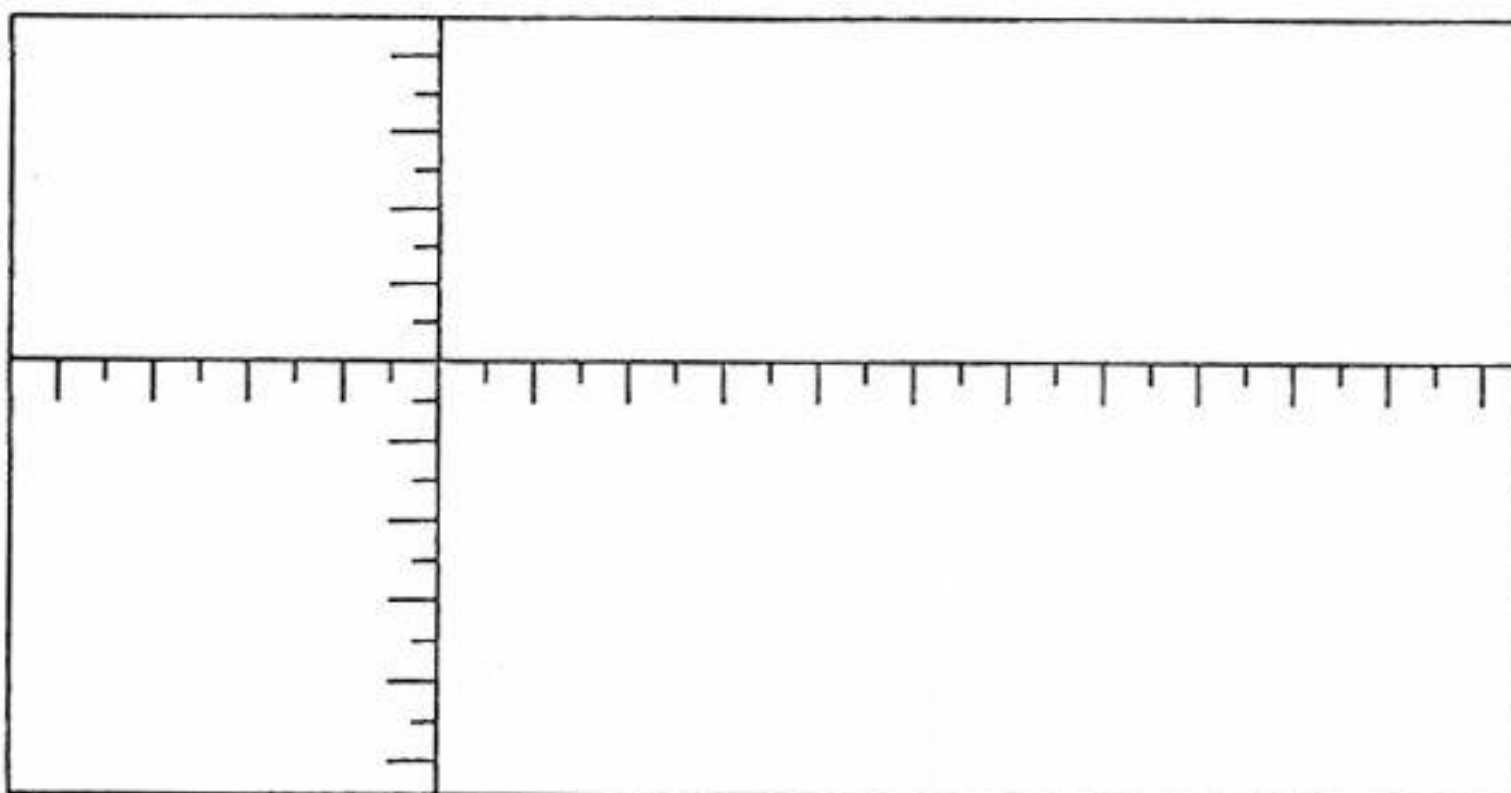
Se si desiderano anche trattini intermedi bisognerà rispondere con un qualsiasi valore diverso da zero alla domanda di riga 210; in caso contrario non verranno tracciati.

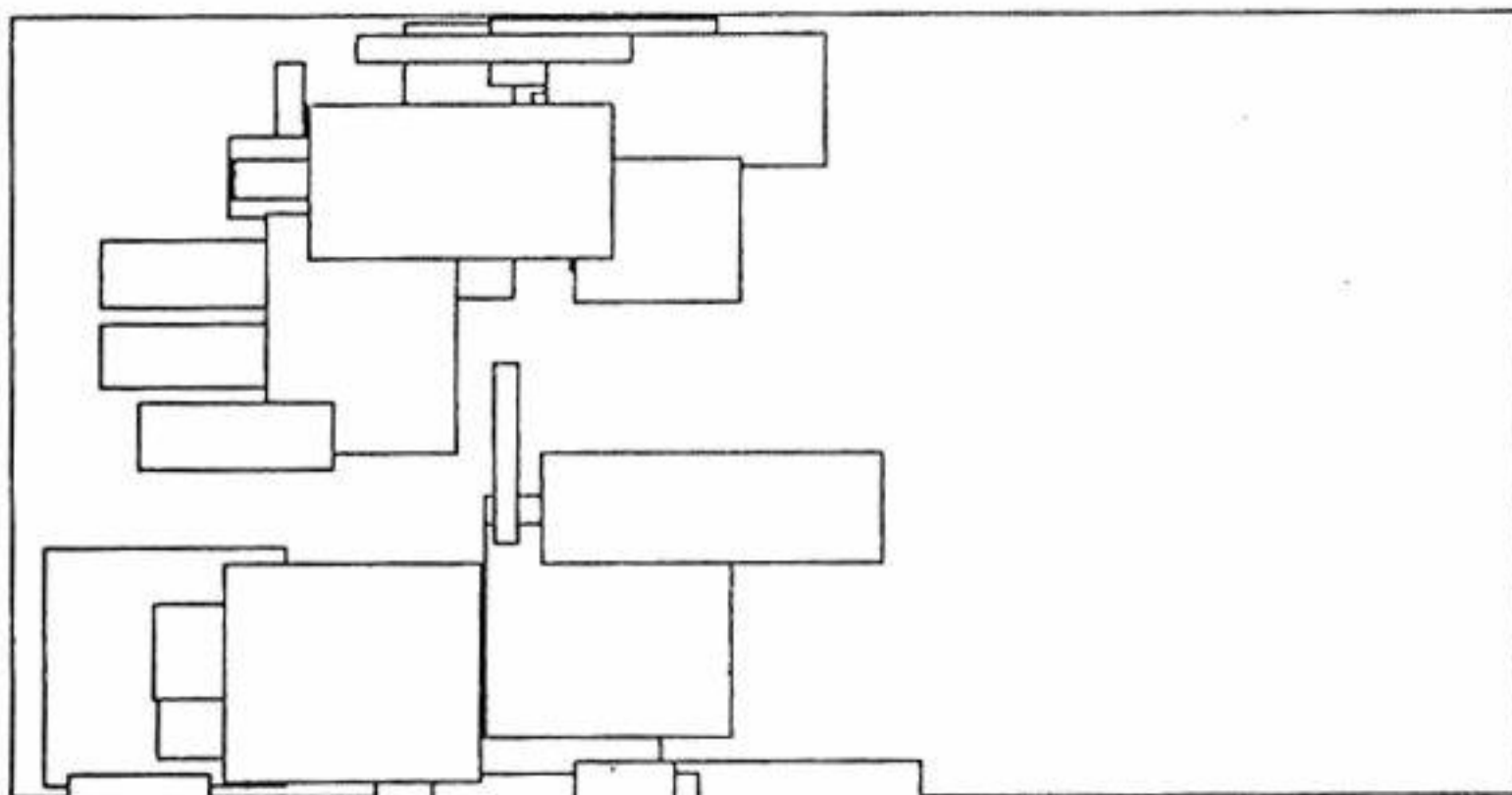
Per "distanza tra i segmenti", ovviamente, si intenderà l'interasse tra i riferimenti.

Si noti che l'origine, qualunque essa sia, rappresenta anche il punto di partenza dei segmenti di riferimento.

Infatti, grazie ai quattro cicli For...Next (righe 250-360) la routine parte dall'origine per tracciare (a destra, a sinistra, in alto ed in basso) i segmenti di riferimento.

Naturalmente sarà cura dell'utente scegliere i valori opportuni dei vari parametri per visualizzare un sistema cartesiano sufficientemente leggibile.

Nel riportare la routine in propri listati, si consiglia di farla elaborare prima di ogni altra. Grazie alla sua elevata velocità, infatti, sarà possibile verificare subito se gli assi tracciati sono quello desiderati e, magari, ripetere l'elaborazione in caso di errore.

```
100 REM DISEGNO ASSI CARTESIANI
110 REM C/16, PLUS/4, C/128
120 :
130 GRAPHIC 0,1
140 PRINTCHR$(19);
150 INPUT"ASSE X (0-199)";X
160 IF X<0 OR X>199 THEN 150
170 INPUT"ASSE Y (0-319)";Y
180 IFY<0ORY>319THEN170
190 INPUT"LUNGH.SEGM.";LS
200 INPUT"DISTANZA SEG";SD
210 INPUT"SEGM.INTERM. (1/0)";S1
220 GRAPHIC1,1:GOSUB 390
230 DRAW 1,0,Y TO 319,Y:REM ASSE X
240 DRAW 1,X,0 TO X,199:REM ASSE Y
250 FORI=Y TO 0 STEP -SD
260 DRAW 1,X,I TO X+LS,I:IFSI=0THEN280
270 DRAW 1,X,I+SD/2 TO X+LS/2,I+SD/2
280 NEXT: FORI=Y TO 199 STEP SD
290 DRAW 1,X,I TO X+LS,I:IFSI=0THEN310
300 DRAW 1,X,I+SD/2 TO X+LS/2,I+SD/2
310 NEXT:FORI=X TO 0 STEP -SD
320 DRAW 1,I,Y TO I,Y-LS:IFSI=0THEN340
330 DRAW 1,I+SD/2,Y TO I+SD/2,Y-LS/2
340 NEXT:FORI=X TO 319 STEP SD
350 DRAW 1,I,Y TO I,Y-LS:IFSI=0THEN370
360 DRAW 1,I+SD/2,Y TO I+SD/2,Y-LS/2
370 NEXT:GETKEYA$:GRAPHIC0:GOTO140
380 REM DISEGNA CORNICE
390 DRAW1,0,0 TO 319,0 TO 319,199
400 DRAW TO 0,199 TO 0,0
410 DRAW1,319,199 TO 0,199 TO 0,0:RETURN
```

```
100 REM CANCELLAZIONE LINEE NASCOSTE
110 REM C/16, PLUS/4, C/128
120 INPUT"RETTANG. NASCOSTI (S/N)";A$
130 IF A$="S" THEN A=1
140 GRAPHIC1,1:GOSUB230
150 Z=120:W=260: GOSUB 210
160 X1=X:Y1=Y:Z=80:W=80:GOSUB210
170 IF A=0 THEN 190
180 BOX0,X1,Y1,X1+X,Y1+Y,,1:REM VUOTO
190 BOX1,X1,Y1,X1+X,Y1+Y,,0:REM BORDO
200 GOTO 150:REM RIPETE CICLO
210 X=INT(RND(0)*Z):Y=INT(RND(0)*W)
220 RETURN
230 REM DISEGNA CORNICE
240 DRAW 1,0,0 TO 319,0 TO 319,199
250 DRAW TO 0,199 TO 0,0: RETURN
```

Linee nascoste

Chi effettua lavori di grafica tridimensionale ha spesso a che fare con un problema classico della grafica computerizzata.

Ci riferiamo alle superfici chiuse che, una volta tracciate, dovrebbero cancellare le linee alle quali si sovrappongono.

Pensando, ad esempio, ad un solido qualunque, come una scatola, è ovvio che le supefici anteriori impediscono la visione di quelle posteriori, a meno che non si abbia a che fare con solidi trasparenti.

Il problema è complicatissimo ed è uno dei maggiori ostacoli incontrati in programmi professionali: è necessario, infatti, determinare esattamente quali linee risultano nascoste (e quindi da cancellare) e quali, invece, devono restare ben visibili.

Si ottiene, tuttavia, una notevole facilitazione se l'elaborazione viene effettuata iniziando il disegno del solido a partire dalla parte posteriore, più lontana dall'osservatore.

In questo modo, infatti, (almeno nei casi più semplici) basterà tracciare la superficie e cancellare qualunque linea sia presente al suo interno. La superficie successivamente elaborata, per come è stata impostata la procedura, "deve" essere sullo stesso piano della precedente oppure (caso più frequente) più vicina all'osservatore.

Possono quindi presentarsi due soli casi: la nuova superficie non "copre" una (o più) superfici precedentemente tracciate: la "cancellazione" del suo interno, pur se inutile, non porta alcun risultato degno di nota. Nel caso in cui, al contrario, alcune linee già visualizzate "cadano" nella nuova figura, queste devono esser cancellate in quanto "coperte" dalla nuova superficie.

La routine pubblicata genera rettangoli casuali "nascosti", o meno, a seconda della indicazione digitata in fase di Input.

La differenza tra le due figure, una relativa all'eliminazione delle linee nascoste e l'altra no, dovrebbe fugare ogni dubbio residuo.

Figure sovrapposte

Il problema delle linee nascoste è risolvibile con una certa facilità grazie all'istruzione specifica che consente di ottenere rettangoli pieni o vuoti.

Nel caso di figure generiche è invece necessario uno studio più attento a causa dell'inevitabile ricorso all'istruzione Paint che porta a vari inconvenienti.

Per riempire una figura disegnata in alta risoluzione, infatti, è necessario individuare, al suo interno, un punto qualsiasi, posizionarvi il cursore ed impartire il comando Paint.

Il computer, a questo punto, verserà la vernice (colore attuale) su tutta la superficie che incontra, individuando, come confine, solo le eventuali linee colorate con lo stesso colore.

Se, ad esempio, è presente sul video solo una circonferenza (che non presenta punti di interruzione), con un comando paint al suo interno otterremo un cerchio; se, invece, impartiamo Paint al suo esterno, verrà colorato l'intero schermo, tranne l'area racchiusa dalla circonferenza; se, poi, questa presenta un "buco" o, comunque, almeno una discontinuità, il comando Paint, riferito all'interno o all'esterno, colorerà l'intero schermo.

E' quindi sufficiente una sola discontinuità perchè Paint, agendo da "riempitore" o da "svuotatore", produca un irreparabile disastro cancellando, o colorando, l'intero schermo ed il suo contenuto.

Per evitare un simile inconveniente è quindi necessario accertarsi di non avere a che fare con disconinuità, e di delimitare con cura la superficie interessata da Paint.

Il programma genera un reticolo sul quale, in seguito, disegna circonferenze casuali (riga 180) che subito si appresta a riempire con Paint (190) perchè non vi è pericolo di fuoruscita di vernice.

A questo punto dovrebbe esser necessario vuotare la stessa area (per risolvere il problema delle linee nascoste) ma, provvedendo subito, si rischia di cancellare porzioni di schermo ben più ampie di quella desiderata.

Si provvede, quindi (riga 200) a disegnare in negativo (cioè, in pratica, a cancellare) una circonferenza (o ellisse) più piccola di un solo pixel che viene quindi riempita del colore opposto (210).

Quindi (220) viene nuovamente disegnata la circonferenza originaria e riempita (230); questa operazione provvede, tra l'altro, anche ad

```
100 REM CIRCONFERENZE SOVRAPPOSTE
110 REM C/16, PLUS/4, C/128
120 :
130 GRAPHIC 1,1: GOSUB 310
140 W=150:Z=280: GOSUB 280
150 X1=X:Y1=Y
160 Z=70:W=70: GOSUB280
170 IFX<1 OR Y<1 THEN160
180 CIRCLE 1,X1,Y1,X,Y:REM CIRCONFER.1
190 PAINT  1,X1,Y1:REM CERCHIO PIENO N.1
200 CIRCLE 0,X1,Y1,X-1,Y-1
210 PAINT  0,X1,Y1 :REM CANCELLAZIONE
220 CIRCLE 1,X1,Y1,X,Y
230 PAINT 1,X1,Y1 :REM SECON.PIENO
240 CIRCLE 0,X1,Y1,X+1,Y+1
250 PAINT 0,X1,Y1 :REM CANCELLAZ.2
260 CIRCLE 1,X1,Y1,X,Y:REM CIRCONF.3
270 GOTO 140: REM RIPETE CICLO
280 X=INT(RND(0)*Z):Y=INT(RND(0)*W)
290 RETURN
300 REM DISEGNA CORNICE
310 DRAW 1,0,0 TO 319,0 TO 319,199
320 DRAW TO 0,199 TO 0,0
330 FORI=0TO319STEP7
340 DRAW1,I,0 TO I,199:NEXT
350 FORI=0TO199STEP7
360 DRAW1,0,I TO 319,I:NEXT:RETURN
```

```
100 REM DISEGNO ASSI CARTESIANI
110 REM C/64 + GW-BASIC
120 :
130 SCREEN 0:PRINTCHR$(147)
140 PRINTCHR$(19);
150 INPUT"ASSE X (0-199)";X
160 IF X<0 OR X>199THEN150
170 INPUT"ASSE Y (0-319)";Y
180 IF Y<0 OR Y>319 THEN 170
190 INPUT"LUNGH.SEGM.";LS
200 INPUT"DISTANZA SEG";SD
210 INPUT"SEGM.INTERM. (1/0)";SI
220 SCREEN1:CLS1: GOSUB390
230 LINE (0,Y)- (319,Y),1:REM ASSE X
240 LINE (X,0)- (X,199),1:REM ASSE Y
250 FOR I=Y TO 0 STEP-SD
260 LINE(X,I)-(X+LS,I),1:IFSI=0THEN280
270 LINE(X,I+SD/2)-(X+LS/2,I+SD/2),1
280 NEXT: FORI=Y TO 199 STEP SD
290 LINE(X,I)-(X+LS,I),1:IFSI=0THEN310
300 LINE (X,I+SD/2)- (X+LS/2,I+SD/2),1
310 NEXT:FORI=X TO 0 STEP -SD
320 LINE (I,Y)-(I,Y-LS)
330 LINE (I+SD/2,Y)-(I+SD/2,Y-LS/2)
340 NEXT:FORI=X TO 319 STEP SD
350 LINE (I,Y)- (I,Y-LS):IFSI=0THEN370
360 LINE (I+SD/2,Y)- (I+SD/2,Y-LS/2)
370 NEXT
372 A$=INKEY$:IFA$=""THEN372
375 SCREEN 0:GOTO140
380 REM DISEGNA CORNICE
390 LINE (0,0)-(319,199),1,B
400 RETURN
```

eliminare altre linee eventualmente ancora presenti.

A questo punto (240) viene "cancellata" la circonferenza più grande di un solo pixel in modo da garantire un confine ben preciso alla successiva operazione (250) di svuotamento seguita, finalmente, dalla definitiva tracciatura della circonferenza desiderata.

Gw-Basic

Come avevamo accennato all'inizio, I listati pubblicati si riferiscono ai computer C/16, Plus/4 e C/128.

Viene, comunque, riportata la versione, per C/64 e Gw-Basic, del listato relativo al disegno degli assi cartesiani.

Non guasta, tuttavia, sottolineare alcune considerazioni:

• Il punto in alto a sinistra dello schermo grafico del C/16, C/128 e Gw-Basic ha coordinate 0,0.

• L'asse X si incrementa verso destra, l'asse Y verso il basso.

• L'istruzione del C/16, Plus/4 e C/128:

Draw Col, X1, Y1 to X2, Y2

traccia (o cancella) una linea di estremi X1,Y1 e X2,Y2 a seconda del valore attribuito al colore (col= 0/1)

In Gw-Basic l'istrzuione che svolge lo stesso compito è la seguente:

Line (X1,Y1) - (X2,Y2), Col

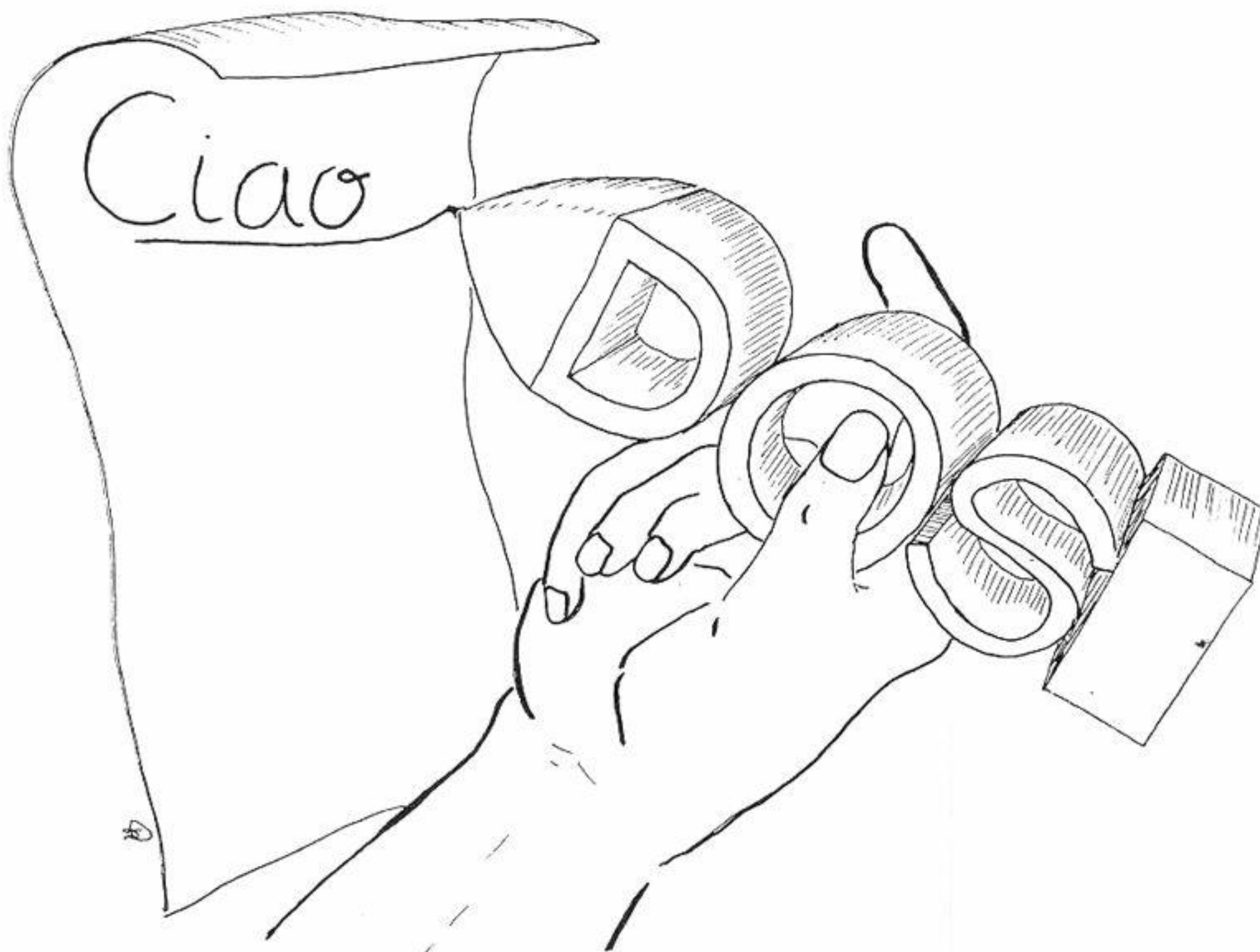
Anche in questo caso Col deve assumere valori 0 e 1.

Per "entrare" in grafica e cacellare lo schermo è necessario, per C/16 e C/128, il comando...

Graphic 1,1

...mentre per il Gw-Basic...

Screen 1: Cls1

Altre analogie sono semplicissime da individuare, soprattutto consultando i libretti di istruzione.

SCRIVERE IN MS-DOS

Tre nuovi comandi esterni da utilizzare in ambiente "Ms-Dos Emulator"

di **Giancarlo Mariani**

A circa due mesi dall'entrata in commercio della seconda versione di Ms-Dos e Gw-Basic per C/64, ecco che inizia la rubrica specifica, che, come promesso a suo tempo, sarà presente su ogni numero di CCC.

Inizieremo subito a parlare di notizie, corredate di programmi utili, che riguardano il simulatore Ms-Dos, lasciando da parte applicazioni per il Gw-Basic sia perchè le sue istruzioni non richiedono particolari spiegazioni, sia, soprattutto, perchè i neo-utenti dimostrano di vler sfruttare al massimo le capacità di emulazione di Ms-Dos.

Segnaliamo subito un paio di (piccoli) bugs indicati dagli stessi lettori:

Il primo riguarda Gw-Basic, e precisamente il comando LLIST, che dovrebbe riportare il listato di un programma su stampante ma, purtroppo, termina l'escuzione tralasciando alcune linee.

Questo problema può essere risolto ricorrendo alla normale procedura del Basic Commodore: per listare dovete digitare...

Open 1,4: Cmd 1: List

...e, alla fine della stampa, concludere con...

Print#1: Close1

Il secondo problema, invece, riguarda Ms-Dos, ed è più o meno dello stesso tipo: il suo comando interno Print, invece di stampare un file su stampante, "inchioda" il procedimento. A questo inconvenienete il presente articolo pone rimedio, presentando un programma che... continuate a leg-

SCHEDA TECNICA

Software applicativo per Ms-Dos Emulator

Hardware richiesto: C/64 oppure C/128; non adattabile ad altri computer Commodore

Uno dei tre comandi pubblicati richiede la disponibilità della stampante MPS-803 o compatibili

E' richiesta la versione 2 del simulatore di Ms-Dos e GW-Basic

Indispensabile il disk drive

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

gere e lo scoprirete.

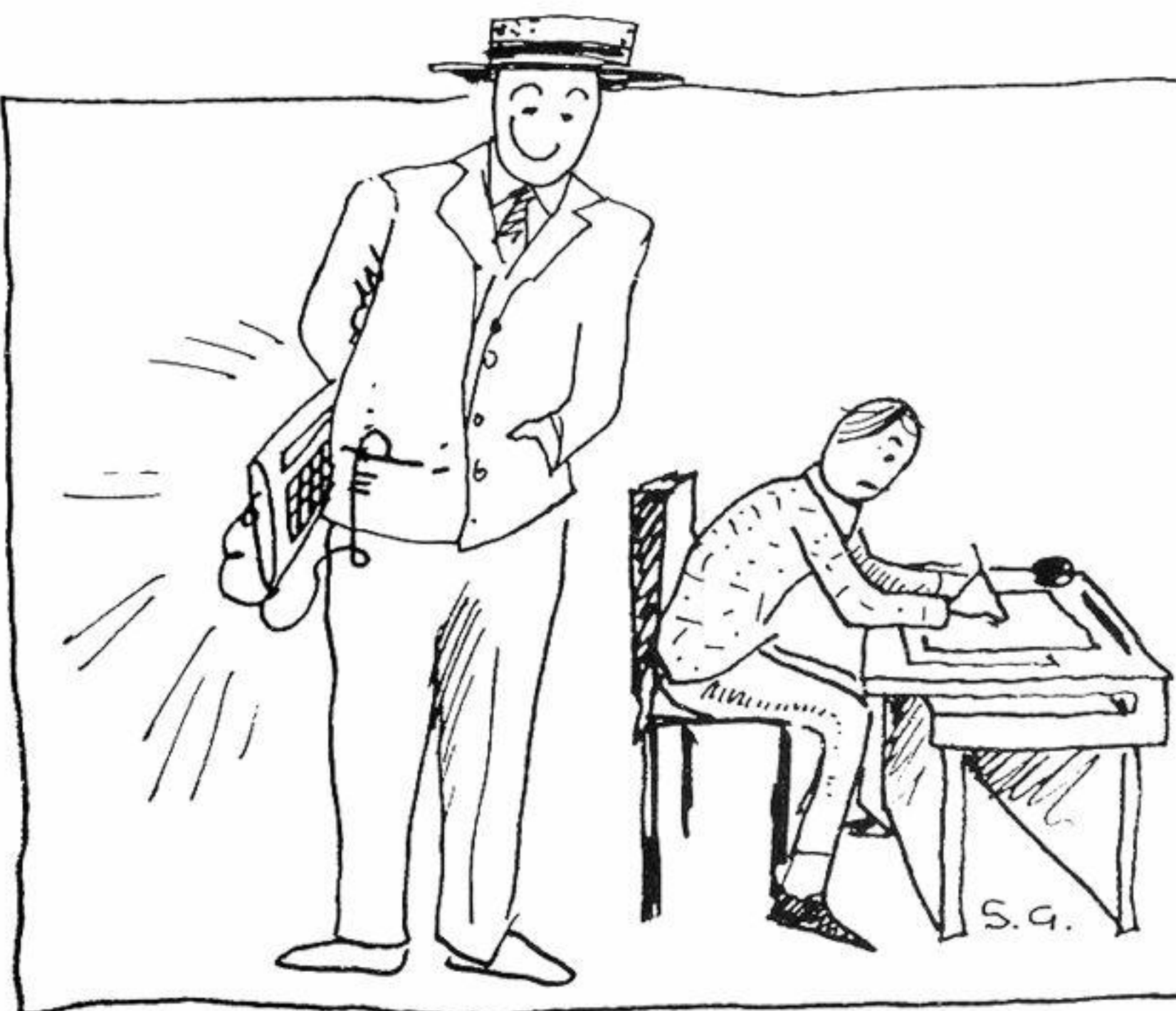
Alcune precisazioni

I bugs scoperti finora sono soltanto quelli indicati in precedenza, di minima importanza; rimane valido, comunque, l'invito rivolto a tutti i lettori che, usando Ms-Dos o Gw-Basic, si accorgano di anomalie, da segnalare prontamente in redazione, in modo da studiare procedure software che siano in grado di correggere gli errori segnalati.

Prima di passare ai programmi proposti, desideriamo chiarire un passo del manuale, che è risultato ambiguo per molti lettori; ci riferiamo alla pagina 10, paragrafo "Comandi".

Vengono infatti indicate alcune aree di memoria Ram che non devono essere alterate da parte di routines L.M. realizzate dall'utente.

Ad una lettura superficiale si potrebbe erroneamente dedurre che Ms-Dos occupa solamente l'area compresa da $9F00 a $9FFF (solo 256 byte!), mentre invece risulta allocato a partire da $0801 (2049) per la lunghezza di circa 25000 bytes.

Pertanto la frase riportata dal manuale si ruferisce ad aree che Ms-Dos usa per funzionare correttamente, OLTRE a quelle che di solito occupa per la sua... esistenza.

L'effettiva area occupata da Ms-Dos, comunque, la potete dedurre leggendo le ultime righe dello stesso paragrafo, in cui si sottolinea che Ms-Dos viene cancellato da programmi che partono da $0801.

Fatte le dovute precisazioni, iniziamo a parlare dei tre programmi da inserire nel dischetto di sistema Ms-Dos e che possono essere utili in svariate occasioni.

I comandi esterni in ambiente Ms-Dos

Il simulatore Ms-Dos consente di trasformare, entro certi limiti, il piccolo computer C/64, dotato di drive, in uno strumento didattico di straordinaria efficacia.

Anche con un C/64, pertanto, è possibile realizzare procedure in accordo con il sistema operativo Ms-Dos "vero".

Uno dei grandi vantaggi di quest'ultimo consiste nello scrivere programmi in linguaggio macchina e richiamarli digitando semplicemente il loro nome.

In questo articolo, appunto, si illustrano ben tre esempi relativi ad altrettanti comandi "esterni" che arricchiranno la vostra collezione di programmi l.m. eseguibili in ambiente Ms-Dos.

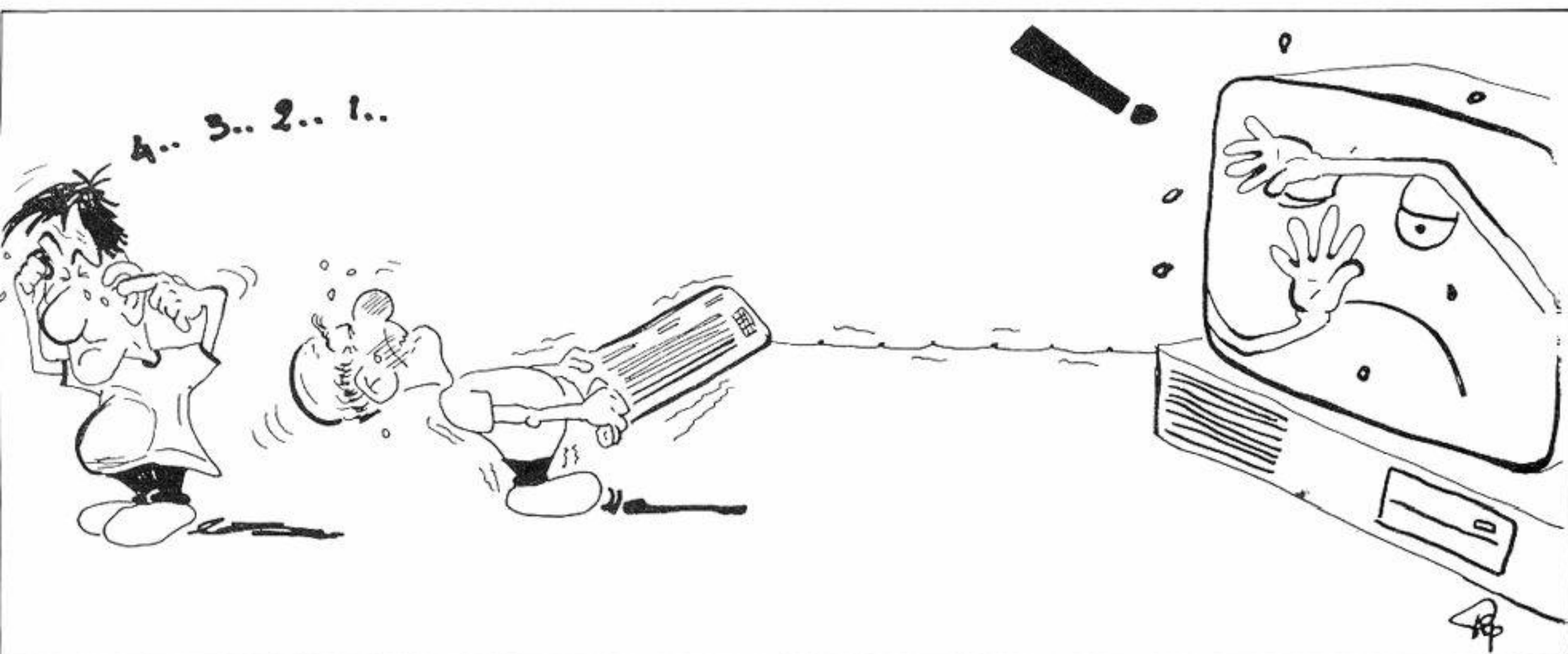
Gli utenti inesperti si limiteranno a digitarli ed utilizzarli; gli altri, cioè gli esperti, avranno la possibilità di incrementare la loro esperienza di programmazione, magari apportando modifiche ai programmi proposti.

Le modalità per creare nuovi programmi sono estremamente semplici: quasi tutti i programmi l.m. che già possedete, infatti, potranno essere trasformati in file di comando validi per l'ambiente Ms-Dos emulator.

Print1

Il primo comando, come detto precedentemente, sostituisce l'inaffidabile Print di Ms-Dos, e gli verrà assegnato il nome di "Print1" per ovvi motivi.

La routine sarà quindi "vista" da Ms-Dos come un comando esterno e, come tale, richiamata specificandone il nome e premendo il tasto Return.

Una volta digitato il comando Print1, apparirà sul video la domanda: "Filename:" a cui si risponderà digitando il nome del file da riportare su stampante. E' da notare che Print1 considera il file SEMPRE partendo dalla directory principale (root), per cui se questo è presente nella root, sarà sufficiente specificare il suo nome, mentre se è

presente in qualche subdirectory sarà necessario specificare l'intero percorso da seguire, come, ad esempio:

a/b/c/nome.ext

Print1 non comprende alcuni controlli, come la pressione di Ctrl-C per fermare la stampa. Si ricorda inoltre che, come intuitivo, la richiesta di stampare file in formato non Ascii può portare a malfunzionamenti di vario tipo.

More

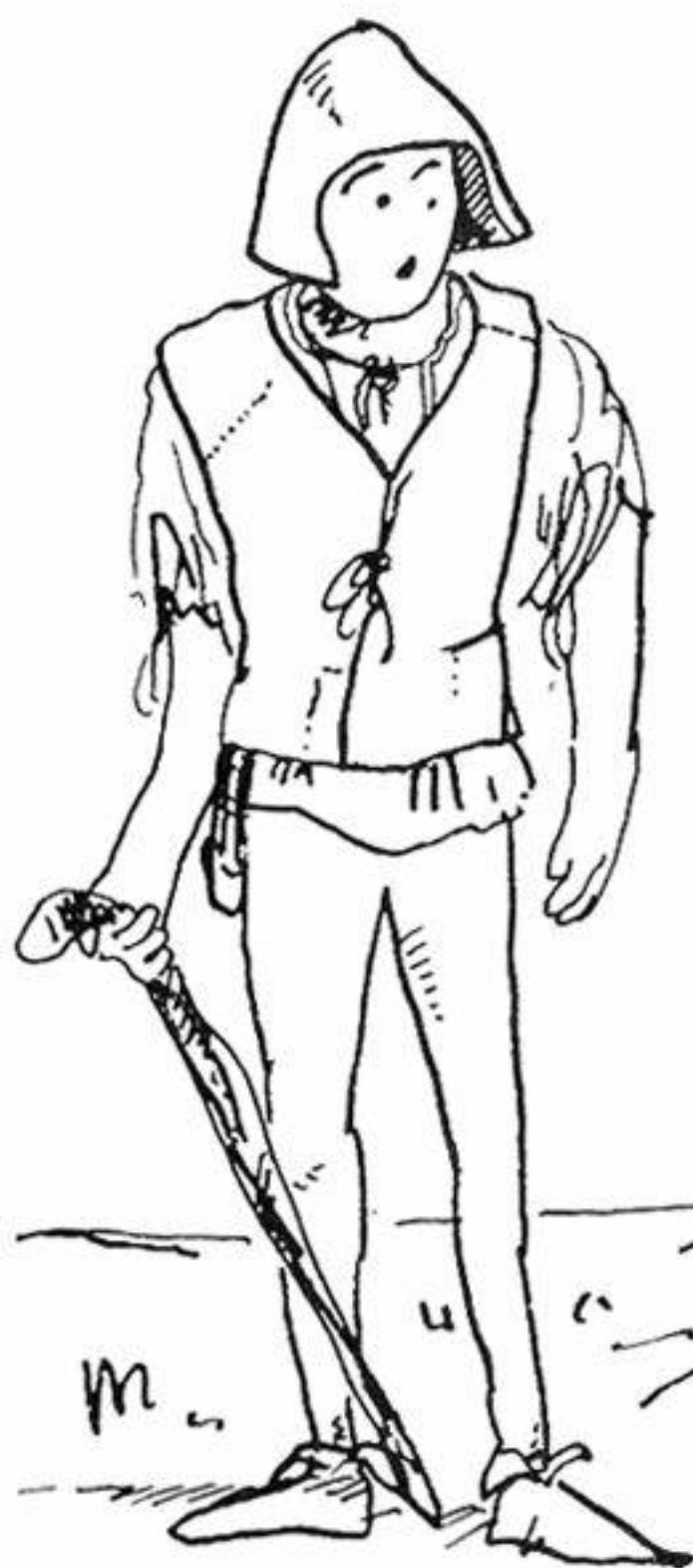
Il secondo comando è un'utilità presente in tutte le versioni Ms-Dos per IBM e compatibili, e viene indicata con: "More".

Questo comando permette visualizzare sullo schermo un file di tipo Ascii, una schermata per volta, in modo che lo scrolling del video non faccia perdere informazioni all'utente. Per richiamarlo è sufficiente digitare MORE (e, ovviamente, premere il tasto Return).

Appena caricato, il comando chiederà il nome del file da visualizzare. Questo andrà inserito secondo le stesse modalità viste per il comando Print1.

Una volta digitato il nome, MORE ricercherà il file su dischetto e, nel caso sia presente, inizierà a visualizzarlo. Al riempimento di una schermata comparirà la scritta "more" e bisognerà premere il tasto Return per continuare la stampa. A differenza di Print1, in questo comando è possibile premere, in qualsiasi momento, i tasti Ctrl-C per fermare la visualizzazione e tornare in ambiente Ms-Dos.

Chkdsk

Il terzo comando simula un'utilità presente nell'Ms-Dos originale, ossia

martoriare il dischetto, e quindi controllare gli eventuali errori. Se chkdsk trova un errore, lo segnala tramite il messaggio "Checking error".

Come trascrivere i comandi

Provvediamo ora a descrivere il modo in cui inserire le tre routines nel disco di sistema Ms-Dos.

Come si vede dai listati pubblicati, vengono memorizzate tramite comandi Read...Data in un programma

non sarà necessario scriverla due volte. Consigliamo dapprima di digitare le suddette righe Data, quindi di salvarle per poterle poi ricaricare ed inserire nei due comandi, uno alla volta. Fate attenzione che in questo caso, se sbagliate a digitare i dati, non funzionerà nessuno dei due comandi.

Come aggiungere comandi

Detto questo, ricordiamo brevemente la procedura per realizzare altri

CHKDSK, che serve per controllare se nel dischetto siano presenti file memorizzati in modo anomalo, ossia che non esistano tracce o settori rovinati. Partendo dall'inizio del disco, il "vero" Chkdsk legge un byte per volta, lo riscrive ed infine lo rilegge per verificare l'eventuale presenza di errori. Alla fine segnala le tracce ed i settori "guasti".

Sarebbe stato molto semplice riprodurre la stessa cosa su Ms-Dos Emulator, ma dal momento che tutti sanno quant'è veloce il 1541, fargli fare un lavoro del genere avrebbe richiesto mezza giornata per dischetto (o quasi...); si è quindi deciso di utilizzare un comando già presente nel Dos del 1541, e precisamente "Validate"; il lavoro che svolge, infatti, è simile a quello richiesto.

Per attivare il comando è sufficiente digitare, da Ms-Dos, CHKDSK. Il C/64 chiederà di inserire il disco da controllare, e quindi di premere il tasto Return. Fatto questo, bisognerà solo aspettare che il 1541 abbia finito di

caricatore Basic: bisognerà quindi prestare la solita attenzione nel digitarlo, dal momento che anche l'errore di un solo dato rischia di inchiodare il computer.

Una volta digitata la routine, registrata e verificata, si deve far partire il programma con il solito RUN ed aspettare che il C/64 abbia finito di leggere i dati. Nel caso in cui non vi siano errori di trascrizione, il computer chiederà di inserire il disco sul quale volete registrare la routine. Fatto questo, basta premere un tasto ed il drive si avvia, salvando il vostro sudato lavoro.

I nomi con cui verranno registrate le routines sono (in accordo con lo standard Ms-Dos) rispettivamente per Print1, More, Chkdisk:

PRINT1.EXE
MORE.EXE
CHKDSK.EXE

La routine "nome" presente sia nel comando More che in Print1 (righe data 1000-1130), è la stessa, e quindi

comandi esterni:

Le routines DEVONO essere in LM, ed essere allocate dall'indirizzo 49152 ($C000) che dovrà anche essere l'indirizzo di partenza della routine (quello della SYS, per intenderci).

Ovviamente queste routines non devono superare l'indirizzo 53247 ($CFFF).

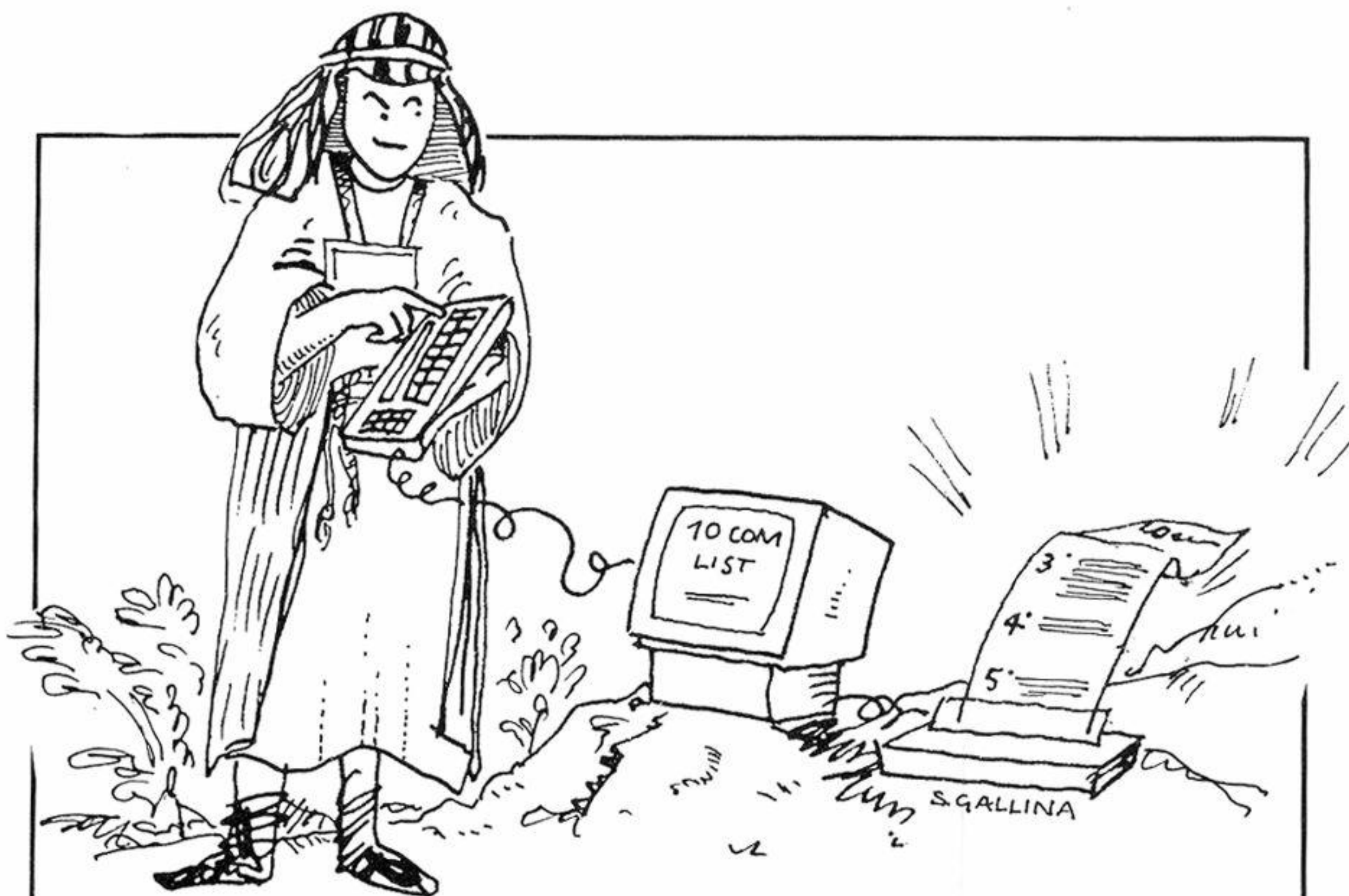
Esse NON devono assolutamente interessare le seguenti porzioni di memoria...

L'area tra $9F00 e $9FFF.

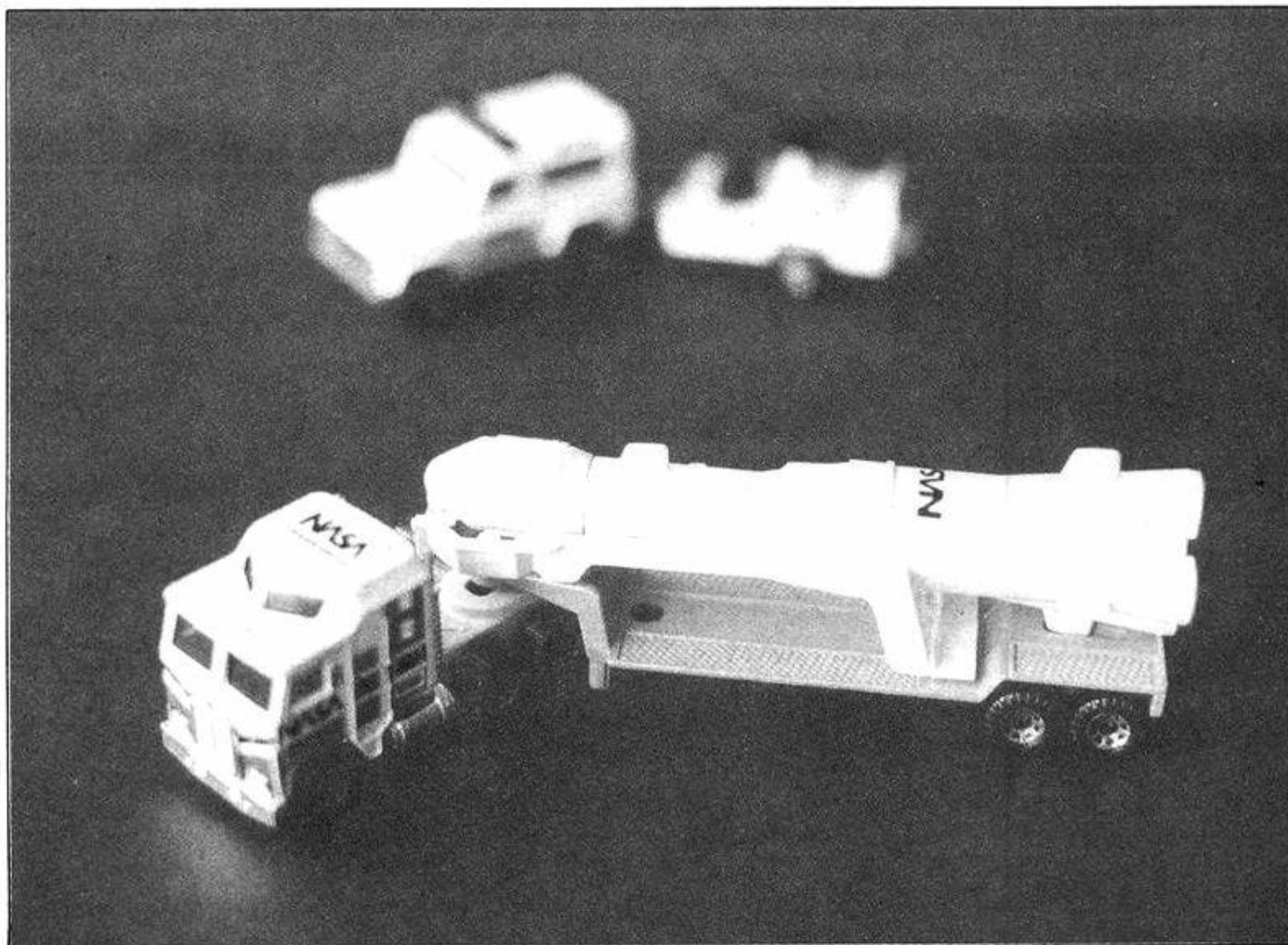
L'area tra $033C e $03FF.

La locazione 2.

...e (lo diciamo di nuovo) i 25000 bytes successivi alla locazione $0800 (2049). Per tornare automaticamente in ambiente Ms-Dos, al termine dell'esecuzione della routine-comando, questa deve terminare con l'istruzione l.m. "Return from subroutine" (RTS).

```
   1 REM  ** 'MORE' (C64+MSDOS E
     MULATOR)
   2 REM  ** BY MARIANI G.
   3 :
 100 FOR  T=0  TO  221:READ  A:P
     OKE  49152+T,A:S=S+A:NEXT
 110 IF  S<>29188  THEN  PRINT"E
     RRORE NEI DATI (ROUTINE NOM
     E)!":END
 120 FOR  T=0  TO  154:READ  A:P
     OKE  49374+T,A:S=S+A:NEXT
 130 IF  S<>51035  THEN  PRINT"E
     RRORE NEI DATI (ROUTINE MOR
     E)!":END
 140 REM  : SALVA SU DISCO
 150 PRINT"[CLEAR][DOWN]INSERISC
     I IL DISCO SUL QUALE VUOI
         SALVARE LA ROUTINE"
 160 POKE 198,0:WAIT 198,1
 170 POKE 43,0:POKE 44,192:POKE
     45,122:POKE 46,193:CLR
 180 SAVE "      MOREEXE",8,1
 190 POKE 43,1:POKE 44,8:POKE 45
     ,115:POKE 46,16:CLR :END
1000 REM  ** ROUTINE NOME **
1005 DATA  076,048,192,013,032,1
     98,073,076,069,078,065,077,
     069,058,000,013,013
1010 DATA  032,194,065,068,032,0
     67,079,077,077,065,078,068,
     032,079,082,032,070
1020 DATA  073,076,069,032,078,0
     65,077,069,013,013,000,000,
     000,000,169,003,160
1030 DATA  192,032,030,171,160,0
     00,032,207,255,201,013,240,
     008,153,208,207,200
1040 DATA  192,047,208,241,192,0
     00,240,228,169,000,153,208,
     207,170,168,141,159
1050 DATA  207,185,208,207,240,0
     20,201,046,240,013,201,047,
     208,005,238,159,207
1060 DATA  208,004,157,160,207,2
     32,200,208,231,169,000,157,
     160,207,170,168,173
1070 DATA  159,207,240,017,201,0
     06,176,088,185,160,207,153,
     000,207,200,232,204
1080 DATA  159,207,144,243,224,0
     05,240,008,169,032,157,000,
     207,232,208,244,185
1090 DATA  160,207,240,007,157,0
     00,207,232,200,208,244,142,
     159,207,169,034,162
1100 DATA  008,160,000,032,186,2
     55,173,159,207,162,000,160,
     207,032,189,255,032
1110 DATA  192,255,162,034,032,1
     98,255,032,207,255,165,144,
     141,158,207,032,204
1120 DATA  255,169,034,032,195,2
     55,173,158,207,240,007,169,
     015,160,192,076,030
1130 DATA  171
1132 REM  ** ROUTINE MORE **
1135 DATA  169,013,032,210,255,1
     69,034,162,008,160,000,032,
     186,255,173,159
1140 DATA  207,162,000,160,207,0
     32,189,255,032,192,255,162,
     034,032,198,255,160
1150 DATA  000,140,255,207,032,2
     07,255,141,254,207,166,144,
     240,013,032,204,255
1160 DATA  169,034,032,195,255,1
     69,013,076,210,255,201,013,
     240,004,192,040,208
1170 DATA  055,160,000,238,255,2
     07,174,255,207,224,020,144,
     043,162,000,142,255
1180 DATA  207,169,111,160,193,0
     32,030,171,169,000,133,198,
     165,197,201,020,208
1190 DATA  009,173,141,002,201,0
     04,208,243,240,192,201,064,
     240,237,201,001,208
1200 DATA  233,169,000,133,198,1
     73,254,207,032,210,255,200,
     166,197,224,020,208
1210 DATA  157,173,141,002,201,0
     04,208,150,240,158,013,060,
     077,079,082,069,062
1220 DATA  013,000,002
```

```
   1 REM  'CHKDSK' (C64+MSDOS EM
     ULATOR)
   2 REM  ** BY MARIANI G.
   3 :
 100 FOR  T=0  TO  156:READ  A:P
     OKE  49152+T,A:S=S+A:NEXT
 110 IF  S<>18745  THEN  PRINT"E
     RRORE NEI DATI":END
 140 REM  : SALVA SU DISCO
 150 PRINT"[CLEAR][DOWN]INSERISC
     I IL DISCO SUL QUALE VUOI
          SALVARE LA ROUTINE"
 160 POKE 198,0:WAIT 198,1
 170 POKE 43,0:POKE 44,192:POKE
     45,158:POKE 46,192:CLR
 180 SAVE "     CHKDSKEXE",8,1
 190 POKE 43,1:POKE 44,8:POKE 45
     ,39:POKE 46,12:CLR :END
 999 REM  ** DATI CHKDSK **
1000 DATA  169,126,160,192,032,0
     30,171,169,000,133,198,165,
     197,201,064,240,250
1010 DATA  201,001,208,242,169,0
     34,162,008,160,015,032,186,
     255,169,001,162,064
1020 DATA  160,160,032,189,255,0
     32,192,255,169,034,032,195,
     255,169,034,162,008
1030 DATA  160,015,032,186,255,1
     69,000,032,189,255,032,192,
     255,162,034,032,198
1040 DATA  255,032,207,255,141,2
     55,207,032,207,255,013,255,
     207,141,255,207,032
1050 DATA  204,255,169,034,032,1
     95,255,173,255,207,201,048,
     208,001,096,169,107
1060 DATA  160,192,076,030,171,0
     13,032,195,072,069,067,075,
     073,078,071,032,069
1070 DATA  082,082,079,082,032,0
     13,000,013,032,201,078,083,
     069,082,084,032,068
1080 DATA  073,083,075,032,065,0
     78,068,032,080,082,069,083,
     083,032,060,195,210
1090 DATA  062,032,000,002
```

```
   1 REM  'PRINT1' (C64+MSDOS EM
       ULATOR)
   2 REM  ** BY MARIANI G.
   3 :
 100 FOR  T=0  TO  221:READ  A:P
       OKE  49152+T,A:S=S+A:NEXT
 110 IF  S<>29188  THEN  PRINT"E
       RRORE NEI DATI (ROUTINE NOM
       E)!":END
 120 FOR  T=0  TO  97:READ  A:PO
       KE  49374+T,A:S=S+A:NEXT
 130 IF  S<>43054  THEN  PRINT"E
       RRORE NEI DATI (ROUTINE PRI
       NT1)!":END
 140 REM  : SALVA SU DISCO
 150 PRINT"[CLEAR][DOWN]INSERISC
       I IL DISCO SUL QUALE VUOI
              SALVARE LA ROUTINE"
 160 POKE 198,0:WAIT 198,1
 170 POKE 43,0:POKE 44,192:POKE
       45,064:POKE 46,193:CLR
 180 SAVE "        PRINT1EXE",8,1
 190 POKE 43,1:POKE 44,8:POKE 45
       ,123:POKE 46,15:CLR :END
1000 REM  ** ROUTINE NOME **
1005 DATA  076,048,192,013,032,1
       98,073,076,069,078,065,077,
       069,058,000,013,013
1010 DATA  032,194,065,068,032,0
       67,079,077,077,065,078,068,
       032,079,082,032,070
1020 DATA  073,076,069,032,078,0
       65,077,069,013,013,000,000,
       000,000,169,003,160
1030 DATA  192,032,030,171,160,0
       00,032,207,255,201,013,240,
       008,153,208,207,200
1040 DATA  192,047,208,241,192,0
       00,240,228,169,000,153,208,
       207,170,168,141,159
1050 DATA  207,185,208,207,240,0
       20,201,046,240,013,201,047,
       208,005,238,159,207
1060 DATA  208,004,157,160,207,2
       32,200,208,231,169,000,157,
       160,207,170,168,173
1070 DATA  159,207,240,017,201,0
       06,176,088,185,160,207,153,
       000,207,200,232,204
```

```
1080 DATA  159,207,144,243,224,0
       05,240,008,169,032,157,000,
       207,232,208,244,185
1090 DATA  160,207,240,007,157,0
       00,207,232,200,208,244,142,
       159,207,169,034,162
1100 DATA  008,160,000,032,186,2
       55,173,159,207,162,000,160,
       207,032,189,255,032
1110 DATA  192,255,162,034,032,1
       98,255,032,207,255,165,144,
       141,158,207,032,204
1120 DATA  255,169,034,032,195,2
       55,173,158,207,240,007,169,
       015,160,192,076,030
1130 DATA  171
1132 REM  ** ROUTINE PRINT1 **
1135 DATA  169,034,162,008,160,0
       00,032,186,255,173,159,207,
       162,000,160,207
1140 DATA  032,189,255,032,192,2
       55,169,035,162,004,160,007,
       032,186,255,169,000
1150 DATA  032,189,255,032,192,2
       55,162,034,032,198,255,164,
       144,240,023,162,035
1160 DATA  032,201,255,169,013,0
       32,210,255,032,204,255,169,
       034,032,195,255,169
1170 DATA  035,076,195,255,032,2
       07,255,164,144,208,226,072,
       162,035,032,201,255
1180 DATA  104,164,144,208,215,0
       32,210,255,032,204,255,024,
       144,197
```

COME SFRATTARE UNA ROUTINE

Spostare un blocco di memoria Ram, all'interno del C/64, presenta problemi risolvibili facilmente

di **Giancarlo Mariani**

Molte volte può capitare che, scrivendo programmi in l.m. o in Basic, si abbia il bisogno di spostare alcuni blocchi di memoria in altre zone; questa esigenza si sente, di solito, in caso di trasferimenti di pagine grafiche, di colori, di dati, di routine l.m. e così via.

Costruire una routine di trasferimento per spostare un determinato blocco di memoria da una zona all'altra, è una cosa semplicissima perfino in Basic (ciclo For..Next).

Definiamo, ad esempio, INIZIO l'indirizzo di partenza del blocco da trasferire, FINE l'indirizzo di fine e DEST l'indirizzo di destinazione (ossia il nuovo indirizzo di partenza del blocco di memoria trasferito). Un programmino Basic, che permetta di svolgere la funzione richiesta, potrebbe essere il seguente:

```
10FORK=INIZIOTOFINE
20POKEDEST,PEEK(K)
25DEST=DEST+1
30NEXTK
```

Per il ciclo for..next non vi sono problemi per nessuno: serve semplicemente a "scandire" i bytes da trasferire, uno alla volta.

La riga 20 è invece quella che effettua materialmente il trasferimento. Nella linea 25 si deve incrementare il valore dell'indirizzo di destinazione, altrimenti si trasferirebbe sempre sulla stessa locazione di memoria.

Il programmino è semplicissimo, ma, come tutti i suoi fratelli (gli altri programmi Basic) ha il difetto di

essere lento, specialmente se il blocco da trasferire è consistente.

Ma ecco che, per l'ennesima volta, viene in aiuto il linguaggio macchina che con la sua potenza e velocità "spazzolerà" in men che non si dica interi blocchi di memoria. Costruire una routine di trasferimento in l.m., è quasi più semplice che in Basic, dal momento che il l.m. è basato soprattutto su trasferimenti di memoria e dati, e quindi il microprocessore possiede alcune istruzioni dedicate allo scopo.

I passi fondamentali per costruire una routine di trasferimento in l.m. sono i seguenti:

- *Leggi l'indirizzo inizio*
- *Leggi l'indirizzo fine*
- *Leggi l'indirizzo destinazione.*
- *Carica un byte da inizio*
- *Trasferiscilo in dest.*
- *Incrementa inizio*
- *Incrementa dest*
- *Confronta inizio con fine: se sono uguali finisci, altrimenti ricomincia dal quarto punto.*

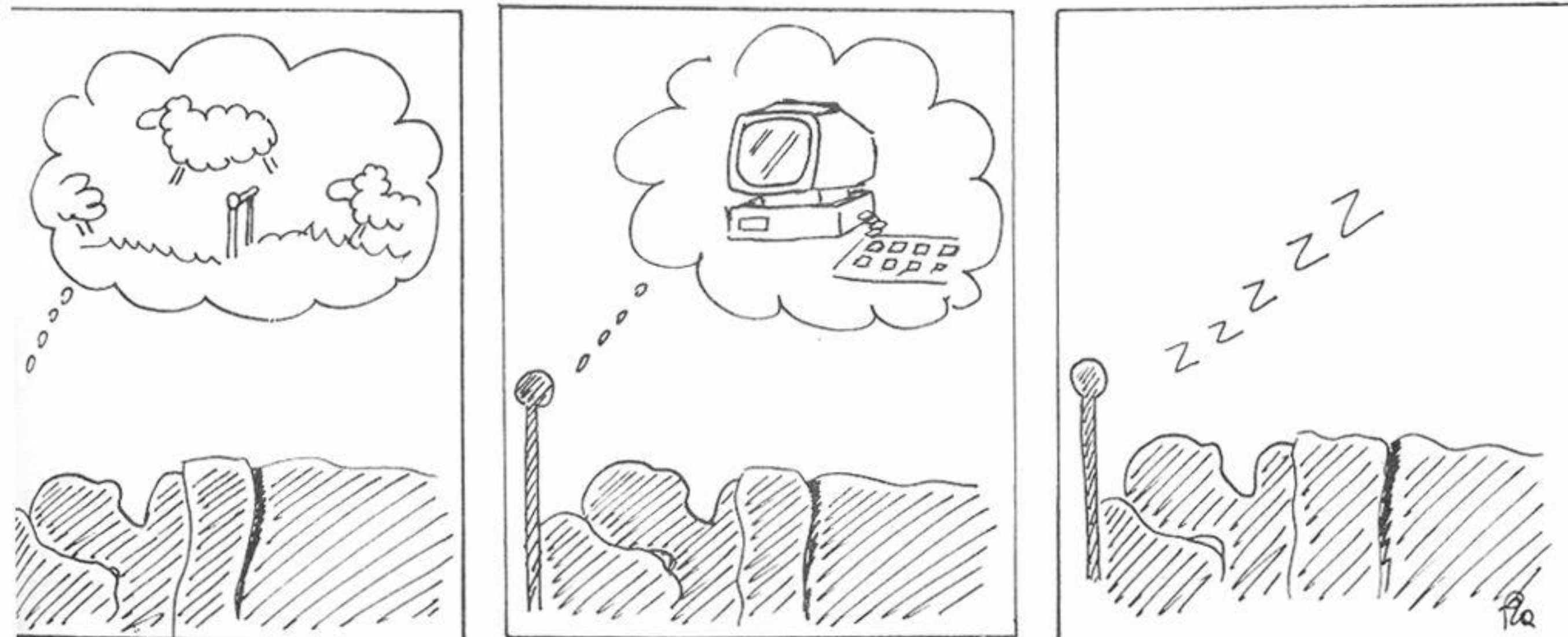
Come si può vedere, non c'è niente di complicato in questi ragionamenti, e osservando l'ultima parte del disassemblato si potrà constatare che compie il lavoro descritto.

Fin qui sembra tutto facile, ed invece, pensandoci bene, sorge una piccola complicazione; va tutto bene finchè le zone di memoria, iniziale e finale, sono sufficientemente distanti fra di loro. Sorgono invece alcuni problemi se si tenta di trasferire, con lo stesso metodo, una zona di memoria all'interno di... se stessa.

Supponiamo, ad esempio, di avere un blocco di memoria che va dalla locazione 5000 alla loc. 5100, e di

volerlo trasferire a partire dalla locazione 5010: seguendo il programma Basic visto in precedenza, iniziamo dal primo byte, cioè 5000, e trasferiamolo a 5010, poi continuiamo spostando 5001 in 5011, poi 5002 in 5012 e così via, fino a 5100 (o no?)

NO!!! perchè quando arriveremo a leggere la locazione 5010, noi leggeremo quello che abbiamo trascritto prima, ossia noi "sporchiamo" la routine prima di trasferirla, e quindi da 5010 sarà presente il contenuto del blocco da 5000, ma solo per 10 bytes; infatti quando arriveremo a leggere la locazione 5010, leggeremo ancora quello che c'era prima a partire da 5000, e così via.

In pratica abbiamo ottenuto una routine formata da

blocchi uguali di 10 bytes ciascuno. Se, poi, volessimo trasferire sempre la stessa routine da 5000 a 5001, otterremmo un risultato ancora più limitato:

Nel primo ciclo sposteremmo il contenuto di 5000 in 5001, nel secondo quello di 5001 (che è diventato ormai uguale a 5000) in 5002, nel terzo 5002 in 5003, e così via, fino ad ottenere una sfilza di bytes tutti uguali, e che non hanno niente a che vedere con la routine di partenza.

Per ovviare a questo "piccolo" inconveniente, esistono due sistemi: il primo consiste nel trasferire all'inizio la routine in una zona di memoria libera da ogni interferenza "esterna", e poi ritrasferirla alla locazione di destinazione. Questo sistema, però, non è molto pratico, dal momento che bisognerebbe sempre disporre di una zona di memoria libera ampia almeno quanto la routine da trasferire, e questo non è sempre possibile.

Un sistema molto più pratico e sbrigativo è quello invece che consiste nel cominciare il trasferimento non dal primo byte, ma dall'ULTIMO.

Infatti, partendo dall'ultimo byte e trasferendo in "avanti", risulta ovvio che non esiste più il problema della sovrapposizione, e tutto funziona a dovere. Il programmino Basic per effettuare questo compito può essere il seguente:

```
10FDEST=(FINE-INIZIO)+DEST
20FORK=FINETOINIZIOSTEP-1
30POKEFDEST,PEEK(K)
40FDEST=FDEST-1
50NEXTK
```

La riga 10 calcola la fine della zona di memoria di destinazione, mentre il ciclo delle righe 20-50 è quello visto prima (al contrario), ossia partendo dall'ultimo byte verso il primo.

Dai ragionamenti visti sinora se ne deduce, inevitabilmente, un altro: se il trasferimento viene effettuato in una zona di memoria più "bassa" di quella di partenza, e si sovrappone a quest'ultima, il senso del trasferimento va di nuovo invertito, altrimenti si avranno gli stessi inconvenienti visti prima.

In pratica, la regola generale è questa:

Se l'indirizzo di destinazione è più basso di quello di partenza, allora il trasferimento va effettuato dal primo byte verso l'ultimo; mentre se è più alto va invertito il senso, ossia dall'ultimo byte al primo.

Ovviamente se l'indirizzo di destinazione è uguale(!) a quello di partenza il senso del trasferimento non è importante.

Trasferisci (27707 - 27823)

La routine di queste pagine è in un certo senso "intelligente", ossia è composta da due trasferimenti, u-

```
1000 PRINT CHR$(147)"ROUTINE DI
     TRASFERIMENTO"
1010 PRINT:PRINT"SYS XXXX,START,
     END,DEST"
1020 PRINT:PRINT"START= INIZIO D
     EL BLOCCO DA TRASFERIRE"
1030 PRINT:PRINT"END= FINE DEL B
     LOCCO DA TRASFERIRE"
1040 PRINT:PRINT"DEST= INDIRIZZO
      DI DESTINAZIONE"
1050 RETURN
1100 DATA 032,253,174,032,138,17
     3,032,247,183,132,250,133,2
     51
1110 DATA 032,253,174,032,138,17
     3,032,247,183,132,252,133,2
     53
1120 DATA 032,253,174,032,138,17
     3,032,247,183,165,021,197,2
     51
1130 DATA 144,045,208,006,165,02
     0,197,250,144,037,165,250,1
     33
1140 DATA 095,165,251,133,096,16
     5,252,133,090,024,101,020,1
     70
1150 DATA 165,253,133,091,101,02
     1,168,138,056,229,250,133,0
     88
1160 DATA 152,229,251,133,089,07
     6,191,163,160,000,177,250,1
     45
1170 DATA 020,230,250,208,002,23
     0,251,230,020,208,002,230,0
     21
1180 DATA 165,251,197,253,208,23
     4,165,250,197,252,208,228,0
     96,-1,17869
```

no dal basso verso l'alto e uno in senso contrario; è in grado di scegliere automaticamente il trasferimento da utilizzare a seconda dell'esito del confronto tra l'indirizzo di partenza e quello di destinazione.

Come routine di trasferimento dall'ultimo byte al primo viene utilizzata una routine già presente nel S.O. del C/64 presente all'indirizzo $A3BF. Essa deve avere come input l'indirizzo di partenza della zona da trasferire nei bytes $5F/$60, l'indirizzo di fine in $5A/$5B e la FINE della zona di destinazione nei bytes $58/$59.

Per il resto, sulla routine non c'è molto da dire poichè è una traduzione dei due programmini Basic visti prima; in ogni caso è possibile studiarla meglio attraverso il disassemblato commentato.

La sintassi è la seguente:

Sysxxxx, inizio, fine, dest

XXXX è, come al solito, l'indirizzo di partenza; questo può essere qualsiasi perchè la routine è rilocabile, come tutte le routine dell'enciclopedia; INIZIO e FINE sono, rispettivamente, le locazioni di inizio e di fine della zona da trasferire, mentre DEST è l'indirizzo di destinazione.

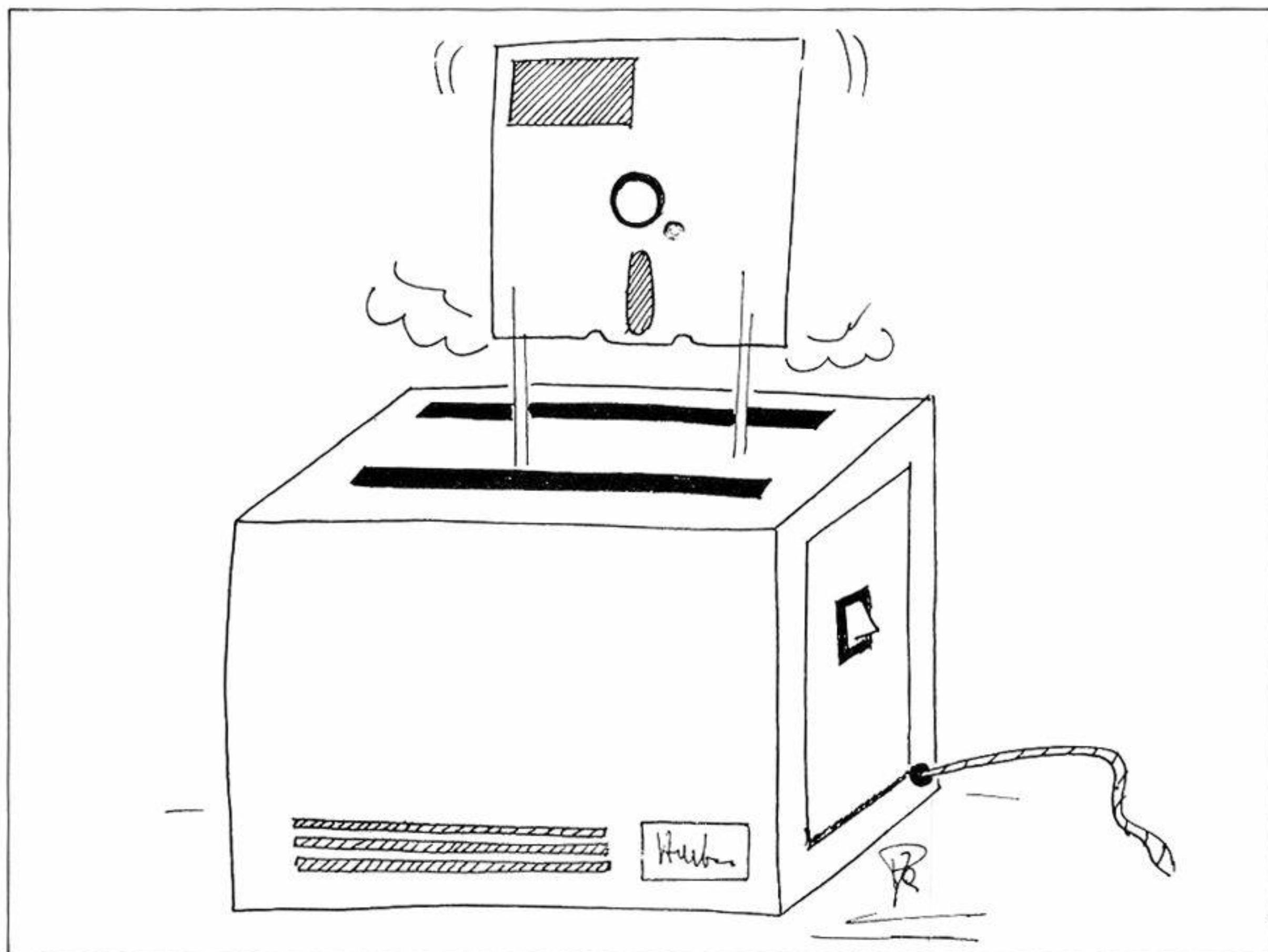
Per chi inizia

Ricordiamo, ai nuovi lettori che ci leggono per la prima volta, che la nostra rivista propone su ogni numero un gruppo di routine in Linguaggio Macchina, per il Commodore 64, interamente rilocabili. Ciò significa che l'utente potrà realizzare una vera e propria enciclopedia in L.M. personalizzata in base alle proprie esigenze.

E' ovvio che le routine proposte, per funzionare adeguatamente, soggiacciono ad alcuni limiti, ben illustrati nei primi numeri di "Nuovo Sistema".

In questa sede ci limitiamo a ricordare che, per evitare malfunzionamenti, è necessario:

- **Fissare il Top di memoria Ram a 20000.**
- **Caricare, servendosi del programma "Caricatore" a suo tempo pubblicato (o analoghi), le routine Basic contenenti le istruzioni Data.**
- **Indicare la prima locazione di memoria in cui si desidera allocare la routine stessa.**
- **Attivare la routine secondo i suggerimenti indicati nei Demo a corredo.**

```
Routine: Transfer
By Mariani Giancarlo

c000 jsr $aefd       ;Contr. virgola.
c003 jsr $ad8a       ;Prende 1' param.
c006 jsr $b7f7       ;(inizio blocco).
c009 sty $fa         ;Lo trasferisce in
c00b sta $fb         ;$FA/$FB.
c00d jsr $aefd       ;Contr. virgola.
c010 jsr $ad8a       ;Prende 2' parametro
c013 jsr $b7f7       ;(fine blocco).
c016 sty $fc         ;Lo trasferisce in
c018 sta $fd         ;$FC/$FD.
c01a jsr $aefd       ;Contr. virgola.
c01d jsr $ad8a       ;Prende 3' parametro
c020 jsr $b7f7       ;(destinazione).
c023 lda $15         ;Confronda dest. con
c025 cmp $fb         ;sorgente: se
c027 bcc $c056       ;dest<sorg salta
c029 bne $c031       ;avanti (trasferim.
c02b lda $14         ;dal basso verso
c02d cmp $fa         ;l'alto).
c02f bcc $c056       ;Altrimenti continua.
c031 lda $fa         ;Trasferisce i vari
c033 sta $5f         ;parametri per la
c035 lda $fb         ;routine S.O. di
c037 sta $60         ;trasferimento ($A3BF)
c039 lda $fc         ;
c03b sta $5a         ;
c03d clc             ;Calcola la fine
c03e adc $14         ;dell'indirizzo di
c040 tax             ;destinazione (per
c041 lda $fd         ;routine S.O. di
c043 sta $5b         ;trasferimento):
c045 adc $15         ;indir.-fine blocco-
c047 tay             ;inizio blocco+
c048 txa             ;destinazione.
c049 sec             ;
c04a sbc $fa         ;
c04c sta $58         ;
c04e tya             ;
c04f sbc $fb         ;
c051 sta $59         ;
c053 jmp $a3bf       ;Trasferim. S.O.
c056 ldy #$00        ;Routine di trasferim.
c058 lda ($fa),y     ;dal basso all'alto.
c05a sta ($14),y     ;Carica e salva byte.
c05c inc $fa         ;increm. inizio blk.
c05e bne $c062       ;
c060 inc $fb         ;
c062 inc $14         ;increm. indirizzo di
c064 bne $c068       ;destinazione.
c066 inc $15         ;
c068 lda $fb         ;Confronta inizio blk
c06a cmp $fd         ;con fine blk: se sono
c06c bne $c058       ;uguali, finisce il
c06e lda $fa         ;trasferimento,
c070 cmp $fc         ;altrimenti continua.
c072 bne $c058       ;
c074 rts             ;
```

UNIRE L'UTILE AL DILETTEVOLE

E' in edicola il nuovo dischetto che contiene ben 15 indispensabili utility (oltre ad un divertente videogame) per gli utenti di C/64-128

di **Michele Maggi**

Il nuovissimo dischetto "Commodore Utilities", racchiude tutte le utility che sono considerate necessarie dai possessori di C/64, più un videogioco molto avvincente per chi, pur utilizzando in modo serio il computer, non vuole rinunciare a qualche momento di svago.

Vediamo ora brevemente le caratteristiche di ogni programma.

TURBOKIT

E' una comoda utility che permette di velocizzare il caricamento dei programmi da disco di circa 5 volte.

Dapprima richiede il nome del programma caricatore (Loader) e successivamente il nome dl programma oggetto (Object) che si desidera caricare in modo turbo.

DATA MAKER

Consente di trasformare una determinata zona di memoria in istruzioni DATA, interessante per chi debba incorporare programmi Assembler all'interno di listati Basic.

La gestione avviene tramite finestre video che garantiscono la semplice utilizzazione.

SHUT OFF

Tipica di sistemi di elaborazione professionali, questa utility, dopo circa 30 secondi di inattività della tastiera, disattiva automaticamente il video.

Ciò va naturalmente a vantaggio della durata dello stesso schermo nonchè di un minor consumo di energia elettrica.

Il video viene automaticamente riacceso non appena viene premuto un tasto qualsiasi.

La routine è allocata nella part alta della memoria in modo da non interferire con altri programmi.

AUTORUN MAKER

Per proteggere i propri programmi da occhi indiscreti, molto spesso è consigliabile dotarli di AUTORUN in modo che il caricamento avvenga con...

Load "Progname",8,1

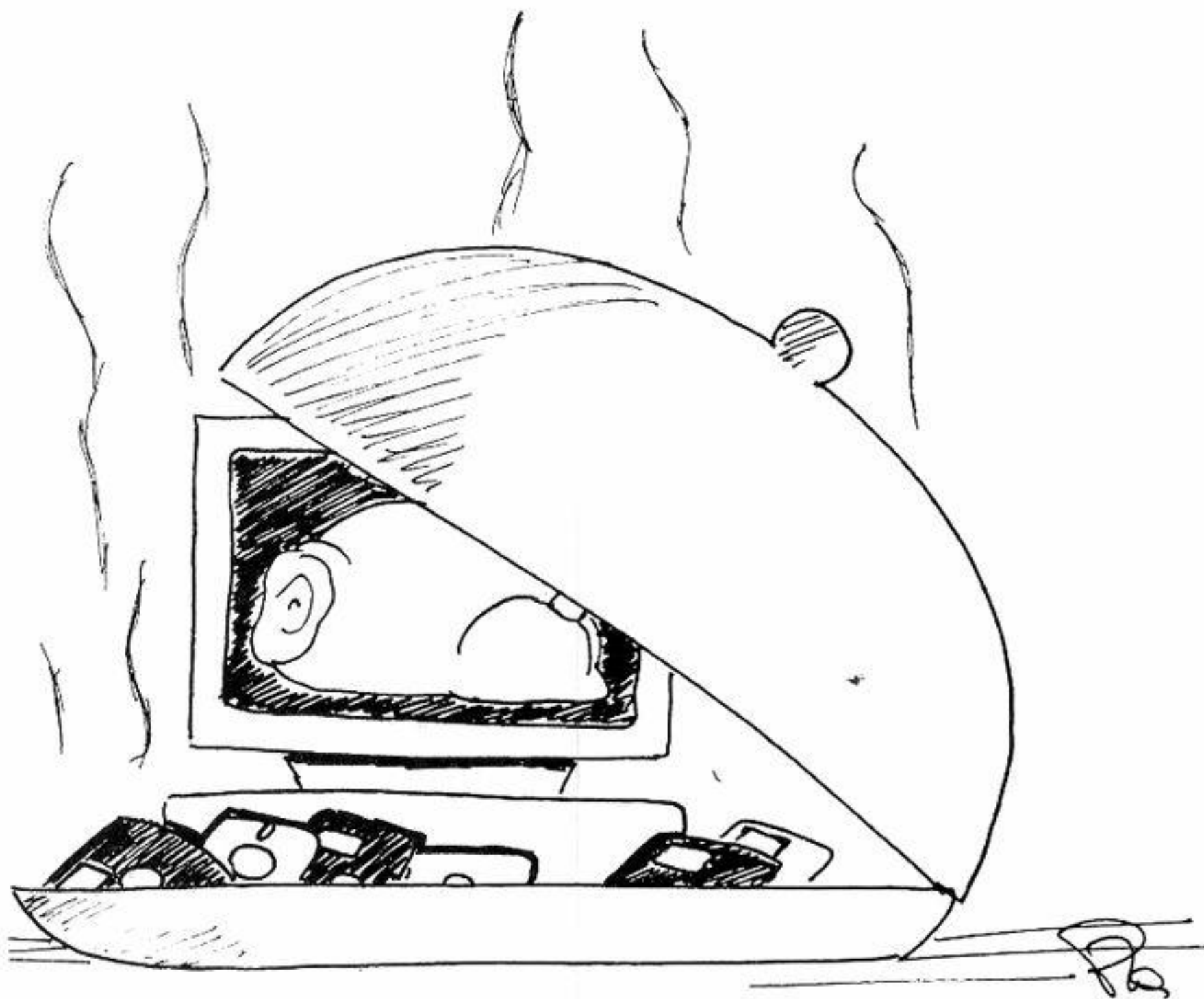
...e si verifichi una esecuzione automatica escludendo la possibilità di essere fermato.

Autorun Maker fa questo lavoro per voi! E' necessario assegnare il nome del programma che si intende dotare di Autorun: il programma farà tutto da solo.

CRUNCHER

Si tratta di un programma che permette di compattare qualsiasi tipo di file, programmi, testi, schermate grafiche, in modo da risparmiare spazio sui dischetti.

Per ricaricare un file compattato è sufficiente caricarlo normalmente ed impartre il Run per lo "scompattamento".

Questo, una volta completato, presenterà in memoria un programma del tutto eguale all'originale, per cui sarà possibile listarlo, modificarlo o mandarlo in esecuzione sia con Run (se in Basic) che con Sys (se in l.m.).

RAM-DISK

Tutti coloro che utilizzano computer evoluti, IBM, Atari, Amiga e così via, hanno certamente utilizzato una Ram-Disk.

La maggior parte degli utenti "comuni", tuttavia, ignora quasi del tutto l'esistenza di questo potente strumento di lavoro.

Che cos'è una Ram-Disk

Si tratta di un'area di memoria RAM protetta da scrittura ed usata a tutti gli effetti come se fosse una periferica (virtualmente, un disk-drive).

I vantaggi della partizione di memoria sono numerosi: in primo luogo, durante il lavoo di programmazione, non sarà necessario salvare su disco, o nastro, successive versioni del programma, con notevole perdita di tempo, ma sarà possibile salvarle temporaneamente in memoria per poi registrare, a fine lavoro, un'unica versione su supporto magnetico.

In secondo luogo, il programma è protetto da qualsiasi evento tranne il caricamento di un programma troppo lungo (superiore ai 110 blocchi).

Sia Run/Stop e Restore che il Reset non alterano il funzionamento del programma per cui il comando specifico (freccia a sinistra + R) è sempre presente e permette di passare dal Basic al modo Ram-Disk.

In terzo luogo funziona da "espansione": infatti, la memoria per il basic è limitata a $7FE0 (circa 30 KB) e lo spazio per la Ram-Disk è di circa 18 KB. In pratica, con l'utility della 4WD-Soft, si dispone i 30+18 KB = 48 KByte (non male!)...

I comandi sono nove e possono essere visualizzati con il comando HELP: load, save, verify, directory, format, rename, erase, help, quit.

DISK MANAGER

Permette di modificare i nomi dei file, cancellarli, recuperarli e proteggerli o sproteggerli dalla cancellazione involontaria.

Le opzioni sono quelle tipiche dei tool di questo genere: unscratch, scratch, unlock.

TYPE

E' un comando di ambiente MS-DOS, molto comodo che permette di esaminare il contenuto di file di testo (SEQ).

Premendo SHIFT si ferma la visualizzazione mentre premendo Run/-Stop si termina.

CROSS REFERENCE

Per il debugging dei programmi Basic è necssario sapere in quali linee vengono elaborate le variabili.

Con il semplice comando SYS 49152, X, dove X rappresenta il numero di colonne con cui vogliamo l'output (12-255), verranno indicati i nomi delle variabili del programma Basic in memoria, affiancati dai numeri di linea in cui sono utilizzate.

DISK MERGER

Con questo programma è possibile fondere due programmi presenti sullo stesso disco in un terzo file che verrà, a sua volta, salvato sul medesimo disco.

RENUMBER

Programmando capita di dover creare spazio tra due linee Basic per inserire una o più istruzioni.

Molto spesso però se la numerazione non lo consente è necessario ricorrere ad utility di Renumber che rinumerano il programma.

La maggior parte di queste tility non aggiornano i GOTO e i GOSUB, con conseguenti malfunzionamenti.

La nostra routine di Renumber non ha invece questo inconveniente.

Le istruzioni per l'uso compaiono sul video alla partenza del programma.

KOALA READER

Molto comodo per chi lavora con il Koala (il diffuso programma grafico) in quanto consente di visualizzare le immagini senza attivare Koala.

ASSEMBLER MASTER

E' formato, in realtà, da tre programmi monitor per linguaggio macchina allocati in diverse zone di memoria.

Il primo parte da 4096 ($1000), il secondo da 32768 ($8000) ed il terzo da 49152 ($C000).

Per un ottimale utilizzo dei monitor è ovviamente necessaria una buona conoscenza dell'Assembler.

FORMAT SAVER

Nella malaugurata ipotesi in cui abbiate formattato involontariamente un disco, contenente file preziosi, questa utility permetterà di recuperarne il contenuto, ripristinando la directory.

Il programma funziona solo in caso di formattazione veloce, cioè quella senza ID.

Una volta attivo, il programma controlla se effettivamente il disco è stato formattato, al fine di evitare di alterare dischi contenenti dati utili.

Successivamente, vengono recuperati gli eventuali files cancellati e, per ognuno di questi, viene inserito come nome, nella Directory, un numero progressivo in quanto è impossibile reperire il nome originario.

Sarà vostra cura caricare i vari files recuperati, analizzarne il contenuto e assegnare loro il nome più appropriato con il comando RENAME.

GRAPH COMPILER

Si tratta di un software di alto livello espressamente dedicato ai patiti della grafica in alta risoluzione.

Permette di gestire grafici tridimensionali ad una notevole velocità, nettamente superiore a quella del Simon's Basic e delle Routine Grafiche di Toma.

L'idea da cui è nato il Graph Compiler era quella di costruire un prodotto di nuova concezione che compilasse anche istruzioni non contenute nel Basic del C/64 (come i comandi DRAW e PLOT).

In un certo senso il Graph Compiler riesce a fare ancora di più: esso infatti compila perfino le istruzioni di notevole utilità e che sono assenti anche nei linguaggi basic più evoluti come quello del C/128; infatti si possono compilare istruzioni completamente inedite (INIT e CDRAW) che permettono di ottenere facilmente schermate di grafici triimensionali con linee nascoste.

DELTAFIGHTER

Si tratta del videogame contenuto, come "omaggio" nella raccolta di utility.

Vi troverete alla guida di un caccia a reazione, armato di missili: la vostra missione è quella di attraversare il sofisticato sistema di difesa dei nemici, addentrandovi il più possibile nel loro territorio, mentre avversari sempre piùagguerriti tenteranno di fermarvi.

Dai primi elicotteri in lento volo, incontrati sul mare ancora prima della costa, fino ai velocissimi jet che compaiono in seguito, tutto vi sarà ostile, con un livello di difficoltà che cresce a mano a mano che il Deltafighter si addentra nelle difese nemiche.

Certamente solo un bravo pilota può portare la missione a compimento; spetta a voi dimostrare di esserlo, raggiungend l'aeroporto nemico e tutte le installazioni che seguono, fino a conquistare il Bunker Corazzato!

Commodore
COMPUTER CLUB
La rivista degli utenti di sistemi Commodore

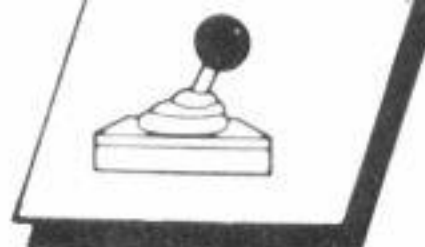
games

From beyond your darkest dreams

15 GIOCHI

Questo mese:

- Quedex
- Clean up service
- Tunnel vision
- Frightmare
- Farstar
- Puppy Love
- Red October
- Soccer King
- Golden path
- Moebius
- Kikstart II
- Sargon III
- The pawn
- Uninvited
- The three stooges

Come leggere le recensioni

Ogni mese, su queste pagine, verranno esaminati e testati i videogame più recenti per i computer Commodore 64 ed Amiga.

Ad ogni descrizione verranno associate una immagine, catturata tra le più belle schermate, ed una breve pagella.

Quest'ultima, pur se, inevitabilmente, frutto di impressioni personali di chi esamina il gioco stesso, ha lo scopo di assegnare una valutazione del livello del software, soprattutto tenendo conto di altri game analoghi disponibili sul mercato.

La pagella, comprende cinque voci:

IMPATTO: indica il livello di interesse suscitato dalla presentazione e dal tema del gioco.

SCENARIO: riguarda l'accuratezza con cui è realizzata la grafica e l'efficacia dei disegni degli sprite.

SUONO: valuta gli effetti sonori presenti e le eventuali musiche di sottofondo.

INTERESSE: si riferisce al livello di interesse che il gioco può suscitare in un giocatore abituato ai videogame e, implicitamente, alla sua probabilità di "permanenza" sui vostri monitor.

TOTALE: ha lo scopo di sintetizzare i precedenti valori con un voto unico.

QUEDEX C/64-128

Un nuovo gioco tipo "Marble Madness", anche se in due dimensioni.

Il videogame è imperniato sulla folle corsa di una pallina alla ricerca dell'uscita del labirinto in cui si trova.

Disseminate per gli schemi di gioco vi sono vari oggetti che possono aiutare il giocatore nella missione.

La grafica, come si conviene a giochi di questo genere, è estremamente curata, mentre gli effetti sonori sono un po' deboli.

Il controllo, molto sensibile, è affidato al joystick tramite il quale dovrete impartire la corretta direzione alla pallina.

Diverse difficoltà vi attendono in corrispondenza dei vari livelli di gioco.

IMPATTO	*8*
SUONO	*6*
SCENARIO	*8*
INTERESSE	*7*
TOTALE	**7**

CLEAN UP SERVICE C/64-128

Anche fare le pulizie può essere pericoloso, molto pericoloso...

Nei panni di un onesto e volenteroso uomo delle pulizie hai l'ingrato e pesante compito di dover rigovernare e pulire una casa enorme.

Nella tua guerra contro la sporcizia dovrai dotarti di tutti gli attrezzi del mestiere che troverai in un apposito box.

E' prevista l'opzione per uno o due giocatori (divertente se si gioca in due).

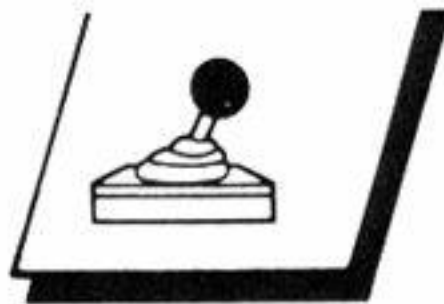
Una piacevole musica di sottofondo tiene compagnia: globalmente un gioco ben riuscito, pur se ricorda vagamente il famoso "Burgertime".

TUNNEL VISION C/64-128

Sul tema di "Pole Position" e "Pitstop II", ecco un nuovo gioco che vi vede impegnati in una corsa che si svolge all'interno di un interminabile tunnel.

E' possibile scegliere tra cinque percorsi differenti ed è addirittura compreso anche un "Tunnel-editor" che permette di costruire il percorso "personale".

La parte grafica è sufficientemente curata mentre il sonoro, a parte una simpatica frase digitalizzata, risulta modesto.

IMPATTO	*7*
SUONO	*6*
SCENARIO	*7*
INTERESSE	*6*
TOTALE	**6**

FRIGHTMARE
C/64-128

Questo gioco vi porterà nella fantastica dimensione dell'incubo.

Durante il vostro viaggio incontrerete tutte le creature che popolano il mondo dell'incubo e dell'immaginario.

Un nuovo gioco con il Terrore (con la T maiuscola) per protagonista, a conferma che questo filone, al cinema come nei videogame, sta raggiungendo un sempre maggiore successo.

Purtoppo la grafica ed il sonoro non sono all'altezza dell'idea del gioco.

Incubi d'oro a tutti!

IMPATTO	*8*
SUONO	*6*
SCENARIO	*7*
INTERESSE	*7*
TOTALE	**7**

FARSTAR
C/64-128

Fa parte della cassetta "Software club" n.18 ed è dedicato a tutti coloro che amano i giochi spaziali.

L'intero gioco si basa sui consueti scontri tra astronavi.

Si svolge su vari livelli il cui grado di difficoltà aumenta in funzione del proseguimento della missione.

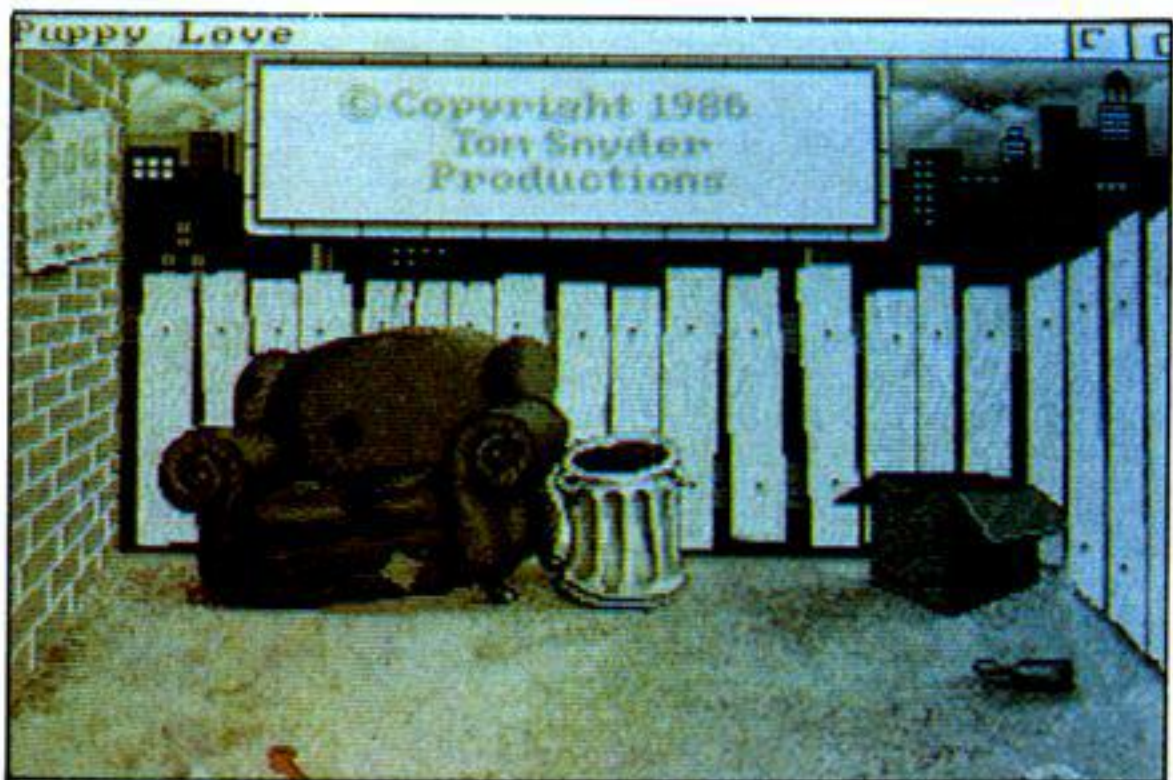
L'intero controllo è affidato al joystick in porta 2.

La parte grafica è molto ben curata ed anche il sonoro fa la sua parte.

PUPPY LOVE
Amiga

Scopo del gioco è addestrare il nostro povero cagnolino ad eseguire tutte le tipiche attività considerate "intelligenti" per una bestia: dare la zampa, rotolarsi, mettersi seduto, eccetera. Una volta addestrato, a suon di ossa e colpi di giornale, si affrontano le "competizioni" con altri possessori di pedigreee raffinati.

Un gioco forse originale; di certo non particolarmente entusiasmante, sia per la grafica di movimento, che per la distorta mentalità, irrispettosa degli animali, che l'ha generato.

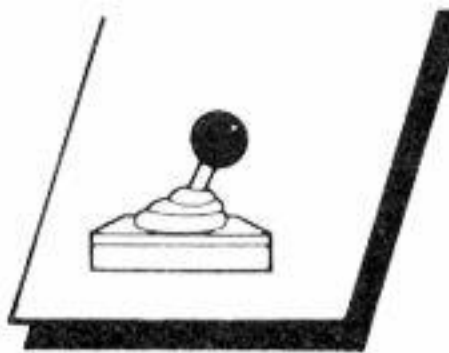
IMPATTO:	*6*
SCENARIO:	*6*
SUONO:	*5*
INTERESSE:	*4*
TOTALE:	**4**

RED OCTOBER
Amiga C/64

Una simulazione di guerra a bordo di un sottomarino, piuttosto dissimile da Silent Service. I controlli sono esclusivamente da mouse, che comporta spesso una lentezza eccessiva. La grafica è più stilizzata, ma più varia e professionale di S.S. Si possono "fare" molte più cose che inseguire le navi nemiche e sparare: tra le altre cose riconoscere le proprie alleate tramite schemi grafici e sonori per evitare di affondarle.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*6*
INTERESSE:	*6*
TOTALE:	**6**

SOCCER KING
Amiga

Si tratta del classico gioco del calcio, per una o due persone, con i giocatori in campo gestiti dai rispettivi joystick. Il programma, tedesco, ha una grafica ovviamente più nitida dell'analogo programma per C/64, ma molto meno precisa. Mancano numerose rifiniture (rimessa con le mani e corner), ma per chi non può fare a meno del football, è l'unica versione esistente per Amiga.

IMPATTO:	*5*
SCENARIO:	*5*
INTERESSE:	*6*
SUONO:	*5*
TOTALE:	**5**

GOLDEN PATH
Amiga

Si manovra un mago in ambiente multischermo, a grafica animata. Tutto viene controllato da mouse e da icone. Il tasto di sinistra fa camminare l'eroe, mentre il tasto destro lo fa lottare. Si tratta di un arcade/adventure di classe. Il numero di schermi da attraversare, di oggetti da raccogliere e di puzzle da risolvere è incalcolabile, la grafica ottima, gli effetti sono buoni, la trama suggestiva.

IMPATTO:	8
SCENARIO:	8
INTERESSE:	8
SUONO:	7
TOTALE:	

MOEBIUS
Amiga C/64

Un misto di avventura ed arcade veramente originale. Nella prima fase ci si allena alla lotta con la spada, ed a giochini di abilità, controllando il gioco con icone o da tastiera. La fase di avventura vera e propria, con uno dei sette personaggi costruiti da noi, ha una grafica più scadente delle fasi di allenamento. Il gioco è complesso ma abbastanza interessante.

IMPATTO:	8
SCENARIO:	7
SUONO:	5
INTERESSE:	6
TOTALE:	6

KIKSTART II
Amiga C/64

Una bellissima gara di motocross, all'altezza dei videogames da sala giochi (ed anche meglio). Infatti è possibile costruire su misura la pista! Si può giocare in due contemporaneamente (su due fette di schermo sovrapposte). I comandi sono controllati dal mouse (icone). Alcune rifiniture potevano essere migliori, ed anche la grafica più fluida (ma è coloratissima).

IMPATTO:	8
SCENARIO:	8
INTERESSE:	8
SUONO:	7
TOTALE:	**8**

SARGON III
Amiga

Uno dei più famosi programmi per giocare a scacchi, disponibile per ogni computer, ora convertito per Amiga. Usa uno dei più efficienti algoritmi di gioco sviluppati su P.C.; purtroppo la grafica, nella versione Amiga, è molto povera (2D), specie se si pensa alle enormi possibilità della macchina.

IMPATTO:	7
SCENARIO:	5
SUONO:	0
INTERESSE:	6
TOTALE:	**6**

THE PAWN
Amiga C/64

Si tratta di una delle prime avventure per Amiga, ma solo da poco è disponibile in versione adatta anche per A/500 e A/2000, tra l'altro notevolmente migliorata (è scritta in C, dunque la manutenzione da parte della Rainbird è facile).

Inoltre è uno dei punti di riferimento per le avventure di Amiga, quindi i nostri lettori "devono" conoscerla.

Si manovra in ambiente medievale, con grafica 3D spettacolare, eventuale sintesi vocale e descrizioni romanzesche (anche per lunghezza!).

Il "parser", cioè la sezione di programma che decodifica le istruzioni, sofisticato ma non perfetto, permette una interazione limitata soprattutto con le altre creature del regno di Kerovnia.

La storia è comunque appassionante e la grafica ottima.

IMPATTO:	7
SCENARIO:	8
SUONO:	0
INTERESSE:	7
TOTALE:	**7**

UNINVITED
Amiga

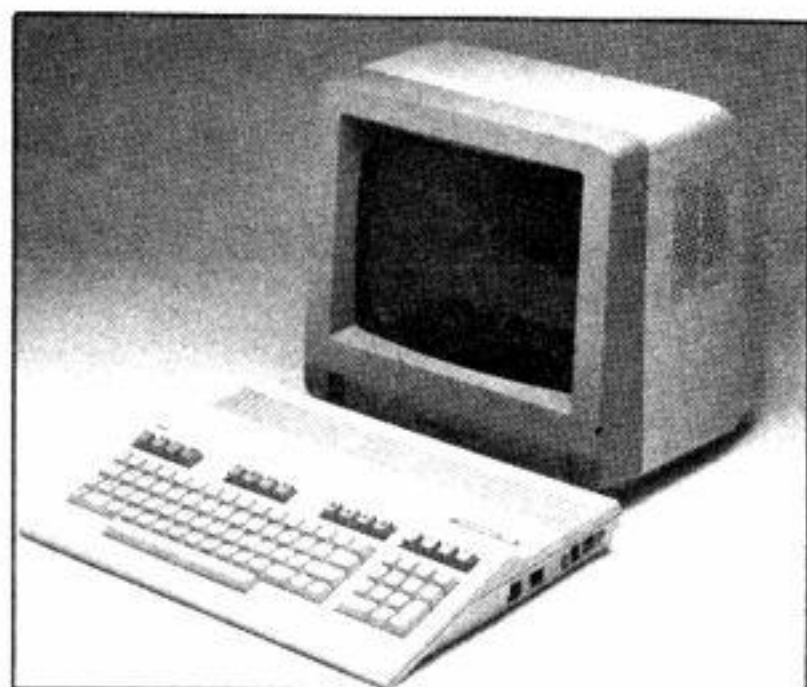
Una delle avventure interattivamente più valide, prodotta dalla competente Mindscape. Sebbene anche qui non si digita mai sulla tastiera.

La trama è da incubo: siamo vittime di un incidente stradale e nostro fratello, quando rinveniamo, è sparito. Per cercarlo finiamo in una casa affollata di spiriti e presenze maligne con tanto di mago che, probabilmente, tiene intrappolato il nostro fratellino.

IMPATTO:	8
SCENARIO:	9
SUONO:	5
INTERESSE:	8
TOTALE:	**7**

THE THREE STOOGES
Amiga

Prodotto dalla Cinemaware, si muovono, come in un film a cartoni animati, tre simpatici scapestrati che debbono raccogliere, in svariati modi (dall'umile lavoro alla truffa) dei soldi per salvare un orfanotrofio. L'interazione col giocatore è migliore che nei precedenti prodotti (SDI, Defender of the Crown), la grafica divertente ed impressionante, il gioco sempre diverso, anche se un po' cervellotico da gestire.

IMPATTO:	9
SCENARIO:	9
INTERESSE:	8
SUONO:	7
TOTALE:	**8**

QUANTO COSTA IL TUO COMMODORE

Codice	Prodotto	Prezzo (IVA esclusa)
A 500	Personal Computer 16/32 BIT CPU 512K RAM incorpora un floppy disk drive da 3" 1/2 da 880Kb e un mouse	L. 950.000
A 501	Espansione di memoria per Amiga 500 512Kbytes e orologio	L. 212.500
A 520/1	Modulatore per Amiga 500 permette di collegare Amiga 500 ad un qualsiasi televisore b/n e colori	L. 40.000
	Cavo collegamento Amiga 500 con TV prescart	L. 27.000
Nuovo C64	Nuovo personal computer CPU 64K RAM computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatore di suono. Collegabile ad un qualsiasi televisore	L. 375.000
1764	Espansione di memoria per C64 256Kbytes	L. 195.000
C 128D	Personal computer CPU 128Kb RAM CPU 128Kbytes espandibile a 512Kbytes, 48 Kbytes ROM, basic 7,0. Tastiera separata. Alta risoluzione grafica, 16 colori + 8 sprites. 40 80 (RGB) colonne. Programmabile in CP/M 3.0. Funzionante in modo C64. Floppy disk da 340Kb incorporato.	L. 895.000
1700	Espansione di memoria per C128. 128Kb	L. 165.000
1750	Espansione di memoria per C128. 512Kb	L. 235.000
1530	Registratore compatibile C64 - C128 -C128D	L. 55.000
1541 II	Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità 170Kbytes in linea. Compatibile con C64	L. 395.000
1571	Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità. 340Kbytes in linea. 410K sotto CP/M. Compatibile con C128 - C128D	L 460.000
1581	Floppy disk drive 3" 1/2. Unità di memoria di massa, drive singolo da 3" 1/2, capacità 800Kbytes. Compatibile C64 - C128 -C128D	L. 420.000
A 1010	Floppy disk drive esterno 3" 1/2	L. 495.000
2080	Monitor a colori alta persistenza alta risoluzione, 14", con audio antiriflesso. Collegabile ad Amiga PC, C64, C128, C128D, 640x400 pixel, 4096 colori	L. 570.000
1084	Monitor a colori. Alta risoluzione, 14", con audio, antiriflesso. Collegabile al C64 - C128 - C128D - Amiga - PC. 640x400 pixel. 4096 colori.	L. 570.000
MPS 1200	Stampante. 80 colonne, 120 cps, bidirezionale. carta in modulo singolo e trascinamento modulo continuo per C64 - C128 -C128D	L. 495.000
MPS 1200 P	Come MPS 1200 per A500	
MPS 1500	Stampante a colori. 80 colonne, 130 cps, 4 colori, bidirezionale, carta in modulo singolo e trascinamento modulo continuo.	*
6499	Adattatore telematico omologato. Collegabile al C64, permette il collegamento a Videotel PGE-Banche Dati.	L. 149.000
1851	Mouse. Per C64 C128D	L. 99.000
1311	Joystick. Comando per giochi	L. 13.500

Codice	Prodotto	Prezzo (IVA esclusa)
Amiga 2000	Microprocessore Motorola 68000; clock 7,16 MHz; 1MB RAM; 256KB ROM; kickstart in ROM. Uscita PAL; 4 uscite a 4 voci su 2 canali (stereofonia). Amiga DOS; Amiga multitasking. 7 slots di sistema (2 combinati Amiga - Ibm PC AT); 1 slot video; 1 accesso diretto CPU 86 pin. Compatibilità MS-DOS interna con schede Janus (XT/AT CARD). Possibilità di un secondo drive interno da 3" 1/2 e di un drive interno da 5" 1/4. Monitor a colori 1084 tastiera; mouse. Workbench; Extras	**L. 2.550.000**
A 1010	Floppy disk drive esterno da 3" 1/2	**L. 495.000**
A 2080	Monitor a colori ad alta risoluzione, a alta persistenza. 14" con audio; 640x400 pixel; 4096 colori	**L. 570.000**
A 1084	Monitor a colori ad alta risoluzione. 14" con audio; antiriflesso; 640x400 pixel; 4096 colori	**L. 570.000**
A 2010	Floppy disk drive interno da 3" 1/2	**L. 280.000**
A 2092	Hard disk da 20MB settorizzabile. PC/Amiga + scheda controller hard disk	**L. 1.085.000**
PC 60/40	Microprocessore 80386 a Clock 8/16 MHz memoria RAM 25/2Kbyte. Hard disk da 40Mb, un floppy disk da 1,2Mb. Sistema operativo MS-DOS 3.2. Scheda EGA di serie. Monitor ADI monocromatico a fosfori verdi da 14". Porta seriale RS232C e parallela Centronics. In opzione: floppy disk drive da 3" 1/2 e 1,44Mb; coprocessore matematico 80387.	
PC 60/80	Stessa configurazione del PC 60/40 ma con un hard disk da 80Mb e, in più, MS-DOS Windows 386	
CP 80387	Coprocessore matematico 80387, 16MHz.	
	Floppy disk drive aggiuntivo, 3" 1/2, 1,44Mb.	
PCI	Microprocessore 8088 (coprocessore 8087 opzionale) frequenza clock 4,77MHz. RAM 512Kb. Espandibile a 640Kb; 1 floppy da 360Kb. Monitor a fosfori verdi 12".	
PC-10-II	Microprocessore Intel 8086 a 16 bit; memoria RAM da 640Kb; memoria ROM (Bios) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.20. Scheda video monocromatico/colore AGA di serie. Due floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C e parallela Centronics.	**L. 1.990.000**
PC 20-II	Microprocessore Intel 8088 a 16 bit; memoria RAM da 640Kb; memoria ROM (BIOS) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.20. Scheda video monocromatico/colore AGA di serie. Un hard disk da 20Mb e un floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C parallela Centronics.	**L. 2.990.000**
PC 40/40	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM da 1Mb. Un hard disk da 40Mb; un floppy disk drive da 1,2Mb. Sistema operativo MS-DOS 3.2. Scheda video monocromatico/colore 132 colonne AGA di serie. Monitor monocromatico a fosfori verdi da 14". Porta seriale RS232 e parallela Centronics. GW Basic	**L. 3.990.000**
CP 80287	Coprocessore matematico 80287, 10MHz	**L. 690.000**
MPS 1200P	Stampante a 9 aghi 120 cps bidirezionale 80 colonne. Interfaccia Commodore per PC 10/III PC 40 PC/60/80	**L. 550.000**
MPS 2010	Monitor a colori ad alta risoluzione 13" con audio	**L. 650.000**
A 2080+ **A 2020**	Scheda Janus XT compatibile + drive Interno da 5" 1/4	**L. 1.210.000**
A 2286 + **A 2020**	Scheda Janus II AT compatibile + drive Interno da 5" 1/4	*
A 2995	Scheda 68020 con processore a 32 bit; 14,28 MHZ	*
A 2997	Interfaccia Genlock multistandard professional	**L. 3.490.000**
A 2998	Interfaccia Genlock VHS/BETA	**L. 1.290.000**
A2999	Framegrabber	*
	Amiga DOS user manual	**L. 26.000**
	Textcraft	**L. 40.000**
	Graphicraft	**L. 40.000**
	Mind Walker	**L. 40.000**
	Pascal	**L. 90.000**
	Lattice C"	**L. 90.000**
	Lisp	**L. 90.000**
	De Luxe Paint	**L. 130.000**
	De Luxe Print	**L. 130.000**
	De Luxe Video	**L. 130.000**
	System five Unix	**L. 1.290.000**
	Superbase in italiano	**L. 130.000**
	Lo Logistix Amiga	**L. 130.000**
	Logistix Versione PC	**L. 740.000**
	Volkswriter 3 in italiano	**L. 745.000**
	Page setter (desk top publishing) in italiano	**L. 180.000**

* *Prezzo da definire*

I COMMODORE POINT

I primi 100 negozi selezionati dalla "Commodore" per la vendita dei suoi prodotti

Negozio	Indirizzo	Località	Telefono
Al Risparmio	*V.le Monza 204*	20128 Milano	Tel. 02-2573440
Braha A. s.d.f.	*Via Pier Capponi 5*	20145 Milano	Tel. 02-437468
E.D.S.	*C.so P.ta Ticinese 4*	20123 Milano	Tel. 02-8322045
Faref	*Via A. Volta 21*	20121 Milano	Tel. 02-650042
Giglioni	*V.le Luigi Sturzo 45*	20154 Milano	Tel. 02-654906
Logitek	*Via Golgi 60*	20133 Milano	Tel. 02-538931
Marcucci	*Via F.lli Bronzetti 37*	20129 Milano	Tel. 02-7386051
Melchioni	*Via P. Colletta 37*	20135 Milano	Tel. 02-57941
Messag. Musicali	*Gal. del Corso*	20100 Milano	Tel. 02-50841
Newel	*Via Mac Mahon 75*	20155 Milano	Tel. 02-323492
Rivola	*Via Vitruvio 43*	20124 Milano	Tel. 02-6694160
F.lli Galimberti	*v. Naz. Giovi 28/36*	20030 Barlassina (Mi)	Tel. 0362-560625
P. Giorgio Ostellari	*Via Molino Arese 65*	20031 Cesano M. (Mi)	Tel. 0362-504392
P. Giorgio Ostellari	*Via Milano 300*	20233 Desio (Mi)	Tel. 0362-621042
GBC Italiana	*V.le Matteotti 66*	20092 Cinisello B. (Mi)	Tel. 02-6181801/6189391
Casa Della Musica	*Via Indipendenza 21*	20093 Cologno M. (Mi)	Tel. 02-2542117
Penati	*Via Verdi 28/30*	20011 Corbetta	Tel. 02-9779401
EPM System	*V.le Italia 12*	20094 Corsico (Mi)	Tel. 02-4407979
Cen. Comp. Pandolfi	*Via Corridoni 18*	20025 Legnano (Mi)	Tel. 0331-546426
Computeam	*Via Vecellio 41*	20035 Lissone (Mi)	Tel. 039-481010
Futura	*Via Solferino 31*	20075 Lodi (Mi)	Tel. 0371-54457
L'Amico del comp.	*V.le Lombardia 17*	20077 Melegnano (Mi)	Tel. 02-9838341
Bit 84	*Via Italia 4*	20052 Monza (Mi)	Tel. 039-320813
I.C.O.	*Via dei Tigli 14*	20090 Opera (Mi)	Tel. 02-5242146
Comif	*Via Autolinee 10*	24100 Bergamo	Tel. 035-218553
Cordani	*Via dei Caniana 8*	24100 Bergamo	Tel. 035-258184
D.R.B.	*Via B. Palazzo 65*	24100 Bergamo	Tel. 035-237292
New Systems	*Via Paglia 36*	24100 Bergamo	Tel. 035-248109
Comp. Team hi-tec	*Via Verdi 1/B*	24030 Carvico (Bg)	Tel. 035-790244
Ott. Opt. Rovetta	*P.zza Garibaldi 6*	24065 Lovere (Bg)	Tel. 035-960705
A.I.S. International	*Via San Carlo 25*	24016 San Pellegrino Terme (Bg)	Tel. 0345-22662
Sisthema	*Via Roma 45*	24067 Sarnico (Bg)	Tel. 035-910750
Computer Center	*Via Cipro 62*	25125 Brescia	Tel. 030-223230
Informatica 2000	*Via Stazione 16/B*	25100 Brescia	Tel. 030-54015
Vigasio Mario	*P.zza Zanardelli 3*	25100 Brescia	Tel. 030-59330/295858
Mister Bit	*Via Mazzini 70*	25043 Breno (Bs)	Tel. 0364-22835
Cavalli Pietro	*Via 10 Giornate 14B*	25030 Castrezzato (Bs)	Tel. 030-714013
Megabyte	*P.zza Duomo 17*	25015 Desenzano d. Garda (Bs)	Tel. 030-9144880
Info Cam	*Provinciale 36*	25050 Gratacasolo (Bs)	Tel. 0364-89379
Il Computer	*Via Indipendenza 90*	22100 Como	Tel. 031-240959
2M Elettronica	*Via Sacco 3*	22100 Como	Tel. 031-278227
Elettrongros	*Via L. Da Vinci 54*	22062 Barzanò (Co)	Tel. 039-957318
Ega	*Via Mazzini 42*	22065 Cassago Br. (Co)	Tel. 039-956307
Ega	*Via Aldo Moro 17*	22043 Galbiate (Co)	Tel. 0341-522028
Data Found Comp.	*Via A. Volta 4*	22036 Erba (Co)	Tel. 031-645761
Righi Elettronica	*Via G. Leopardi 26*	22077 Olgiate C. (Co)	Tel. 031-946766
Fumagalli	*Via Cairoli 48*	22053 Lecco (Co)	Tel. 0341-863341
Cima Eelettronica	*Via L. Da Vinci 7*	22053 Lecco (Co)	Tel. 0341-371106
Mondo Computer	*Via Giuseppina 11/B*	26100 Cremona	Tel. 0372-882079
Prisma	*Via B. Da Dovara 8*	26100 Cremona	Tel. 0372-437900
Elcom/GBC	*Via IV Novem. 56/58*	26013 Crema (Cr)	Tel. 0373-83393
Euroelettronica	*Via XX Settembre*	26013 Crema (Cr)	Tel. 0373-86966
Computer	*Galleria Ferri 7*	46100 Mantova	Tel. 0376-325616
32 Bit (Comp. St.)	*Via C. Battisti 14*	46100 Mantova	Tel. 0376-326770
Log. Inf. Comp.	*V.le M. Grappa 32*	27029 Vigevano (Pv)	Tel. 0381-81883
M. Visentin	*C.so V. Emanuele 76*	27029 Vigevano	Tel. 0381-83833
Computer Line	*Via G. Carducci 4*	29100 Piacenza	Tel. 0523-30691
Delta Computer	*Via Mar. Resisten. 15/4*	29100 Piacenza	Tel. 0523-753318
Sover	*Via IV Novembre 60*	29100 Piacenza	Tel. 0523-34388
Fotonova		23010 S. Pietro di Berbenno (So)	Tel. 0342-492319
Il Centro Elettr.	*Via Morazzone 2*	21100 Varese	Tel. 0332-231006
Busto Bit	*Via Gavinana 17*	21052 Busto A. (Va)	Tel. 0331-625034
Crespi G. & C.	*V.le Lombardia 59*	21053 Castellanza (Va)	Tel. 0331-503023
Computer Shop	*Via A. Da Brescia 2*	21013 Gallarate (Va)	Tel. 0331-798612
J.A.C. Nuo. Tec.	*Via Matteotti 38*	21018 Sesto C. (Va)	Tel. 0331-923134
Bit Micro	*Via Mazzini 102*	15100 Alessandria	Tel. 0131-443252
West Records	*C.so Roma 85*	15100 Alessandria	Tel. 0131-441090
S.G.E. Elettronica	*Via Bandello 19*	15057 Tortona (Al)	Tel. 0131-867709
Rossi Computers	*C.so Nizza 42*	12100 Cuneo	Tel. 0171-2339
Punto Bit	*C.so Langhe 26/C*	12051 Alba (Cn)	Tel. 0173-49833
Curetti Augusto	*C.so Italia 3*	12054 Mondovi (Cn)	Tel. 0174-42014
Ditta Elettrogamma	*c.so Risorgim.to 20*	28100 Novara	Tel. 0321-176358
Elcom	*C.so Mazzini 11*	28100 Novara	Tel. 0321-391293
Programma 3	*V.le Buonarroti 8*	28100 Novara	Tel. 0321-36367/399903
Punto Video	*c.so Risorgim.to 391*	28100 Novara	Tel. 0321-477367
Computer	*Via Monte Zeda 4*	28041 Arona (No)	
All Computer	*C.so Garibaldi 106*	28021 Borgomanero (No)	Tel. 0322-844142
Micrologic	*Via Giovanni XIII 2*	28037 Domodossola (No)	
Elliott Comp. Shop	*Via Don Minzoni 32*	28044 Intra (No)	Tel. 0323-43517
ABA Elettronica	*Via C. Fossati 5/P*	10100 Torino	Tel. 011-302065
Alex Comp.&giochi	*C.so Francia 333/4*	10142 Torino	Tel. 011-7730184
C.D.M. Elettronica	*Via Marochetti 17*	10126 Torino	Tel. 011-636345/634900
Computing New	*Via Marco Polo 40/E*	10129 Torino	Tel. 011-501512
De Bug	*c.so V. Emanuele II 22*	10100 Torino	Tel. 011-832986
Desme Universal	*Via San Secondo 95*	10100 Torino	Tel. 011-592551
F.D.S.	*Via Borgaro 86/D*	10100 Torino	Tel. 011-2168900
MT Informatica	*C.so G. Cesare 58*	10100 Torino	Tel. 011-850955/238803
New Bus. Comp.	*Via Nizza 45*	10100 Torino	Tel. 011-214235
Radio Tv Mirafiori	*C.so Un. Sov.ca 381*	10135 Torino	Tel. 011-616190/6197189
SMT Elettronica	*Via Bibiana 83/B*	10147 Torino	Tel. 011-218243
Paul e Chico V.	*Via V. Emanuele 52*	10023 Chieri (To)	Tel. 011-9470295
Bit Informatica	*Via V. Emanuele 154*	10073 Cirié (To)	Tel. 011-9205455
Eurex	*C.so Indipendenza 5*	10086 Rivarolo C.se (To)	Tel. 0124-27984
Ditta Elettrogamma	*C.so Bormida*	13100 Vercelli	Tel. 0161-53689
Elettronica	*Strada Torino 15*	13100 Vercelli	Tel. 0161-393163
C.S.I. Teorema	*Via Losana 9*	13051 Biella (Vc)	Tel. 015-28622
St. Fotogr. Imarisio	*P.zza M. Libertà 7*	13039 Trino (Vc)	Tel. 0161-82081
F.lli Gatti	*Via Festaz 75*	11100 Aosta	Tel. 0165-35659

A chi chiedere aiuto se il tuo Commodore ha problemi

ABRUZZO

Audio Computer
Via Umbria, 7
65100 Pescara
Tel. 085/29.33.75

CALABRIA

Service Center
Via Parisio, 25
87100 Cosenza
Tel. 0984/75.74.1

CAMPANIA

Electrical Engineer
Via Supportico Lopez, 5/A
80137 Napoli
Tel. 081/44.44.44 - 29.34.08

Marvin Service
Via Nazionale delle Puglie, 344
80013 Casalnuovo (Na)
Tel. 081/84.24.22.4

EMILIA ROMAGNA

Centro Riparatori
Via Lenin, 48/I
41012 Carpi
Tel. 059/64.07.70

Centro Riparatori
Via Galvani, 4
41100 Modena
Tel. 059/21.66.02

Dr. Sax
Via D. Creti, 77/C
40128 Bologna
Tel. 051/35.25.39

Maser
Via Collegio di Spagna, 10
40064 Ozzano Emilia (Bo)
Tel. 051/79.84.48

Centro Raccolta
Via Di Corticella, 177
40128 Bologna

S.I.R.A. srl
Via Aniene, 43/45
48100 Ravenna
Tel. 0544/64.22.3

FRIULI VENEZIA GIULIA

Audio Video Service
Via Gemelli, 9
33170 Pordenone
Tel. 0434/57.11.04

ET Italia
Via Tavagnacco, 89
33100 Udine
Tel. 0432/48.13.39

Hitech
Via Nordio, 9
34100 Trieste
Tel. 040/74.11.89

LAZIO

Computer Service Italia
Via Sebino, 49
00199 Roma
Tel. 06/85.03.86

Tecnicomp
Via dei Georgofili, 65
00159 Roma EUR
Tel. 06/51.33.73.9

Elettrotel srl
Via Verolengo, 20
00167 Roma
Tel. 06/62.31.06.8

LIGURIA

Siragusa
Via Milano, 41
16126 Genova
Tel. 010/26.16.55

Teleradio
Via XXV Aprile, 70
18100 Imperia
Tel. 0183/21.96.2

LOMBARDIA

Abiservice srl
Via Ponale, 48
20100 Milano
Tel. 02/64.37.49.6

Catme
Via Severoli, 9
20147 Milano
Tel. 02/41.52.96.2

Computer Lab
Viale Monte Nero, 66
20135 Milano
Tel. 02/54.64.43.6

Computer Service
Via Genala, 19/B
26100 Cremona
Tel. 0372/43.58.61

MARCHE

Car
Via Bruno Buozzi, 18
60020 Ancona
Tel. 071/80.44.88

PIEMONTE

Grun A.R.
Via De Sanctis, 126/F
10142 Torino
Tel. 011/70.72.47.2

Computer Service Torino
Via Reiss Romoli, 122/11
10100 Torino
Tel. 011/22.02.66.6

PUGLIA

Viga
Via Domenico Morea, 42
70124 Bari
Tel. 080/41.37.06

SARDEGNA

Alacram Technologies
Via Livenza, 3
09123 Cagliari
Tel. 070/28.72.38

SICILIA

Cat Elettronica
Via Ravenna, 7/A
95100 Catania
Tel. 095/43.86.70

Co.As. Informatica
Via Raffaele Mondini, 3
90143 Palermo
Tel. 091/29.52.09

TOSCANA

G.L.V. Elettronica
Via Pietrasantina, 113
56100 Pisa
Tel. 050/56.20.35

Paolo Paolieri
Via Perfetti Ricasoli, 70
50100 Firenze
Tel. 055/43.61.72.0

TRENTINO ALTO ADIGE

Elecomp
Via Druso, 52/A
39100 Bolzano
Tel. 0471/42.12.8

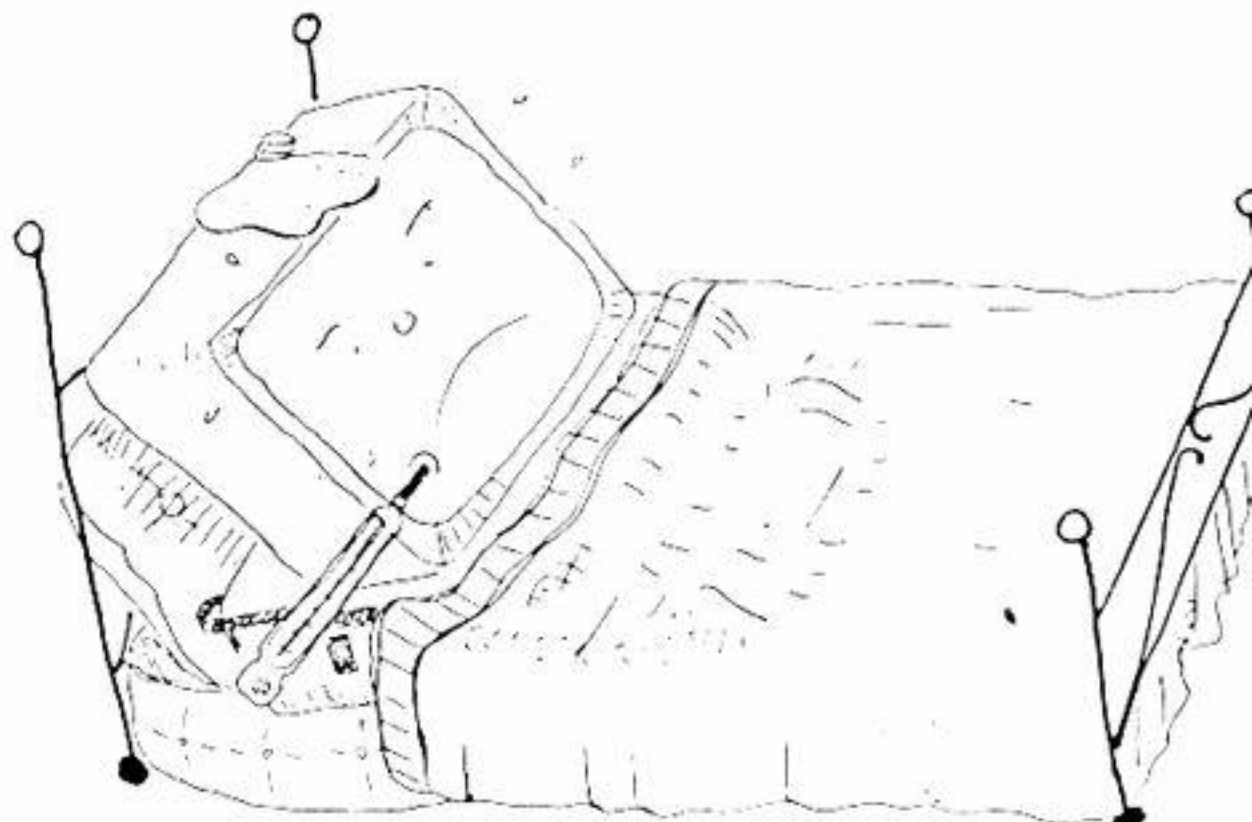
UMBRIA

H.C.H.
Via Ruggero D'Andreotto, 31/A
06100 Perugia
Tel. 075/75.33.53

VENETO

Carpanese Elettronica Telecomunicazioni
Strada VII Martiri, 101
35100 Padova
Tel. 049/62.41.60

Sistel
Via Decorati al Valor Civile, 67
30100 Mestre
Tel. 041/93.53.32

Pesente Giovanni
Via Pitagora, 6
37100 Verona
Tel. 045/56.59.8

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N.13, 14, 15 della rivista "Commodore" e sui N.1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L.100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.45000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.70000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.